aws

Guida per gli sviluppatori

Amazon Cognito

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

Amazon Cognito: Guida per gli sviluppatori

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Amazon Cognito? 1
Bacini d'utenza 2
Pool di identità 3
Caratteristiche di Amazon Cognito 4
Bacini d'utenza 4
Pool di identità 6
Confronto tra pool di utenti e pool di identità di Amazon Cognito 8
Nozioni di base su Amazon Cognito 12
Disponibilità regionale 13
Prezzi di Amazon Cognito 13
Come funziona l'autenticazione 13
Autenticazione SDK 14
Autenticazione UI ospitata 17
Autenticazione tramite provider di identità di terzi 20
Autenticazione del pool di identità 23
Termini di Amazon Cognito 26
Generali 27
Bacini d'utenza 29
Pool di identità 31
Lavorare con AWS gli SDK 32
Iniziare con AWS 33
Registrati per un Account AWS 33
Crea un utente con accesso amministrativo 34
Nozioni di base sui bacini d'utenza 36
Esempio di React SPA 36
Creazione di un'applicazione 41
Crea un ambiente per sviluppatori Lightsail 42
Esempio di app mobile Flutter 43
Creazione di un'applicazione 48
Passaggi successivi 50
Creazione di un pool di utenti 50
Aggiungi un client di app UI ospitato 55
Aggiunta di un provider del social 58
Aggiunta di un provider SAML 66

Nozioni di base sui pool di identità 69
Creazione di un pool di identità in Amazon Cognito 69
Configurazione di un SDK 71
Integrazione dei provider di identità 71
Ottenere le credenziali 72
Opzioni introduttive aggiuntive 73
Integrazione con app 75
Autenticazione con AWS Amplify 76
Creazione di un'interfaccia utente con Amplify 77
Autenticazione con SDK AWS 78
Autorizzazione con Amazon Verified Permissions 78
Autorizzazione API con autorizzazioni verificate 80
Esempio di policy per un utente Amazon Cognito. 83
Esempi di codice 86
Amazon Cognito Identity 87
Azioni 88
Esempi di servizi incrociati 110
Provider di identità Amazon Cognito 112
Azioni 121
Scenari 238
Amazon Cognito Sync 363
Azioni 363
Best practice per le applicazioni multi-tenant 366
Pool di utenti per tenant 368
Client di app per tenant 370
Gruppi di pool di utenti per tenant 372
Attributi personalizzati per tenant 374
Suggerimenti per la sicurezza del multi-tenancy 376
Scenari comuni di Amazon Cognito 378
Autenticazione con un bacino d'utenza 378
Accesso alle risorse lato server 379
Accesso alle risorse con API Gateway e Lambda 380
Accedi ai AWS servizi con un pool di utenti e un pool di identità 381
Autenticazione con terze parti e accesso ai servizi AWS con un pool di identità 381
Accedi alle AWS AppSync risorse con Amazon Cognito 382
Bacini d'utenza di Amazon Cognito 384

Funzionalità 385
Registrazione 385
Accesso 386
Interfaccia utente ospitata 387
Sicurezza 387
Esperienza utente personalizzata 388
Monitoraggio e analisi 388
Integrazione dei pool di identità di Amazon Cognito 389
Autenticazione 389
Flusso di autenticazione del bacino d'utenza 392
Client dell'app 402
Utilizzo dei dispositivi 413
Utilizzo dell'API e degli endpoint 419
Autenticazione API del pool di utenti 422
Aggiornamento di un pool di utenti 430
Configurazione SMS 432
Aggiornamento di un pool di utenti con un AWS SDK o un' AWS CDK API REST 432
Interfaccia utente ospitata e server OAuth 434
Configurazione dell'interfaccia utente ospitata con AWS Amplify 435
Configurazione dell'interfaccia utente ospitata con la console Amazon Cognito 436
Visualizzazione della pagina di accesso 439
Dettagli sull'interfaccia utente ospitata dei pool di utenti di Amazon Cognito 440
Configurazione di un dominio 442
Personalizzazione delle pagine Web integrate 452
Come utilizzare l'interfaccia utente ospitata 459
Ambiti e server di risorse 477
Autorizzazione M achine-to-machine (M2M) 478
Informazioni sugli ambiti 479
Informazioni sui server di risorse 480
Aggiunta di un accesso tramite terze parti 485
Funzionamento dell'accesso federato nei pool di utenti di Amazon Cognito 485
Responsabilità di un'app come provider di servizi con Amazon Cognito 487
Informazioni importanti sull'accesso di terze parti ai pool di utenti di Amazon Cognito 487
Provider di identità 488
Fornitori di identità sociali 495
Provider SAML 503

Fornitori OIDC ... 535
Specificazione mappature degli attributi ... 545
Collegamento di utenti federati a un profilo utente esistente ... 550
Utilizzo di trigger Lambda ... 554
Considerazioni importanti ... 556
Aggiunta di un trigger al bacino d'utenza ... 559
Evento trigger Lambda per il bacino d'utenza ... 560
Parametri comuni del trigger Lambda del bacino d'utenza ... 560
Origini dei trigger Lambda in base all'evento ... 561
Origini dei trigger Lambda per funzione ... 568
Trigger Lambda di pre-registrazione ... 571
Trigger Lambda di post-conferma ... 581
Trigger Lambda di pre-autenticazione ... 585
Trigger Lambda di post-autenticazione ... 589
Trigger Lambda di richieste ... 594
Trigger Lambda di pre-generazione del token ... 608
Trigger Lambda di migrazione utenti ... 628
Trigger Lambda di messaggi personalizzati ... 634
Trigger Lambda del mittente personalizzato ... 642
Utilizzo dell'analisi dei dati di Amazon Pinpoint ... 659
Ricerca delle mappature regioni Amazon Cognito e Amazon Pinpoint ... 660
Integrazione di app con Amazon Pinpoint ... 664
Analisi ... 665
Gestione degli utenti ... 667
Consentire la registrazione dell’utente ... 667
Registrazione e conferma degli account utente ... 670
Creazione di utenti come amministratore ... 696
Aggiunta di gruppi a un bacino d'utenza ... 702
Gestione e ricerca degli utenti ... 705
Recupero degli account utente ... 709
Importazione di utenti in un bacino d'utenza ... 710
Attributi ... 728
Requisiti password ... 742
Impostazioni e-mail ... 744
Configurazione e-mail predefinita ... 745
Configurazione e-mail di Amazon SES ... 746

Configurazione dell'account e-mail ... 752
Impostazioni per i messaggi SMS ... 758
Prima impostazione degli SMS nei bacini d'utenza di Amazon Cognito ... 760
Utilizzo di token ... 767
Utilizzo di token ID ... 769
Utilizzo del token di accesso ... 773
Utilizzo del token di aggiornamento ... 777
Revoca dei token ... 779
Verifica di un JSON Web Token ... 781
Caching dei token ... 787
Accesso alle risorse dopo aver effettuato l'accesso ... 790
Accesso alle risorse con autorizzazioni verificate ... 379
Accesso alle risorse con API Gateway e AWS AppSync ... 793
Accesso alle AWS risorse utilizzando un pool di identità ... 795
Utilizzo delle caratteristiche di sicurezza ... 800
Aggiunta di MFA ... 801
Aggiunta di sicurezza avanzata ... 813
AWS WAF ACL Web ... 830
Distinzione tra lettere maiuscole e minuscole ... 835
Deletion protection (Protezione da eliminazione) ... 836
Gestione delle informazioni utente ... 838
Pool di identità di Amazon Cognito ... 845
Utilizzo dei pool di identità ... 847
Ruoli IAM dell'utente ... 849
Identità autenticate e non autenticate ... 849
Attivazione o disattivazione dell'accesso guest ... 849
Modifica del ruolo associato a un tipo di identità ... 850
Modifica dei provider di identità ... 851
Eliminazione di un pool di identità ... 853
Eliminazione di un'identità da un pool di identità ... 854
Utilizzo di Amazon Cognito Sync con pool di identità ... 854
Concetti del pool di identità ... 857
Flusso di autenticazione dei pool di identità ... 858
Ruoli IAM ... 868
Attendibilità del ruolo e autorizzazioni ... 882
Best practice di sicurezza ... 884

Le migliori pratiche di configurazione IAM ... 884
Best practice per la configurazione del pool di identità ... 886
Utilizzo di attributi per il controllo degli accessi ... 888
Utilizzo degli attributi per il controllo dell'accesso con i pool di identità di Amazon Cognito ... 889
Esempio di utilizzo di attributi per la policy di controllo degli accessi ... 891
Disattivazione di attributi per il controllo degli accessi ... 893
Mappature di provider predefinite ... 893
Utilizzo del controllo degli accessi basato su ruoli ... 895
Creazione dei ruoli per la mappatura dei ruoli ... 895
Concessione delle autorizzazioni per il passaggio di ruoli ... 896
Utilizzo dei token per l'assegnazione dei ruoli agli utenti ... 897
Utilizzo di una mappatura basata su regole per assegnare i ruoli agli utenti ... 898
Attestazioni dei token da utilizzare nella mappatura basata su regole ... 900
Best practice per il controllo accessi basato sui ruoli ... 901
Ottenere le credenziali ... 902
Accesso ai servizi AWS ... 909
Provider di identità esterni con pool di identità ... 912
Facebook ... 912
Login with Amazon ... 921
Google ... 926
Accedi con Apple ... 939
Provider Open ID Connect ... 946
Provider di identità SAML ... 950
Identità autenticate dagli sviluppatori ... 954
Informazioni sul flusso di autenticazione ... 954
Definisci un nome del provider per sviluppatori e associalo con un pool di identità ... 955
Implementa un provider di identità ... 955
Aggiornamento della mappa degli accessi (solo Android e iOS) ... 964
Ottenimento di un token (lato server) ... 965
Connessione a un'identità social esistente ... 966
Supporto delle transizioni tra provider ... 967
Cambio delle identità ... 971
Android ... 971
iOS - Objective-C ... 971
iOS - Swift ... 972
JavaScript ... 972

Unità ... 973
Xamarin ... 974
Amazon Cognito Sync ... 975
Nozioni di base su Amazon Cognito Sync ... 976
Configurazione di un pool di identità in Amazon Cognito ... 976
Archiviazione e sincronizzazione dei dati ... 976
Sincronizzazione dei dati ... 976
Inizializzazione del client di Amazon Cognito Sync ... 977
Comprendere i set di dati ... 979
Lettura e scrittura dei dati nei set di dati ... 981
Sincronizzazione dei dati locali con lo store di sincronizzazione ... 983
Gestione dei callback ... 987
Android ... 987
iOS - Objective-C ... 989
iOS - Swift ... 993
JavaScript ... 996
Unità ... 999
Xamarin ... 1002
Sincronizzazione push ... 1004
Creazione di un'app Amazon Simple Notification Service (Amazon SNS) ... 1005
Abilitazione della sincronizzazione push nella console di Amazon Cognito ... 1005
Uso della sincronizzazione push nella tua app: Android ... 1006
Uso della sincronizzazione push nella tua app: iOS - Objective-C ... 1008
Uso della sincronizzazione push nella tua app: iOS - Swift ... 1011
Amazon Cognito Streams ... 1014
Amazon Cognito Events ... 1017
Utilizzo della console Amazon Cognito ... 1022
La console del pool di utenti ... 1023
La console dei pool di identità ... 1025
Sicurezza ... 1027
Protezione dei dati ... 1028
Crittografia dei dati ... 1028
Gestione dell'identità e degli accessi ... 1029
Destinatari ... 1030
Autenticazione con identità ... 1031
Gestione dell'accesso con policy ... 1034

Funzionamento di Amazon Cognito con IAM ... 1037
Esempi di policy basate su identità ... 1047
Risoluzione dei problemi ... 1051
Uso di ruoli collegati ai servizi ... 1054
Registrazione di log e monitoraggio ... 1058
Monitoraggio dei costi ... 1059
Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas ... 1061
Registrazione delle chiamate all'API Amazon Cognito con AWS CloudTrail ... 1077
Convalida della conformità ... 1104
Resilienza ... 1105
Considerazioni sui dati regionali ... 1105
Sicurezza dell'infrastruttura ... 1106
Analisi della configurazione e delle vulnerabilità ... 1106
AWS politiche gestite ... 1107
Aggiornamenti alle policy ... 1108
Assegnazione di tag alle risorse ... 1111
Risorse supportate ... 1111
Limitazioni applicate ai tag ... 1112
Gestione dei tag con la console ... 1112
Esempi di AWS CLI ... 1112
Assegnazione di tag ... 1113
Visualizzazione dei tag ... 1114
Rimozione dei tag ... 1114
Applicazione di tag durante la creazione delle risorse ... 1115
Operazioni dell'API ... 1116
Operazioni API per i tag del bacino d'utenza ... 1116
Operazioni API per i tag del pool di identità ... 1116
Quote ... 1117
Informazioni sulle quote di frequenza di richiesta API ... 1117
Categorizzazione delle quote ... 1117
Operazioni API del bacino d'utenza di Amazon Cognito con una gestione speciale della frequenza delle richieste ... 1118
Monthly active users (Utenti attivi mensili) ... 1119
Gestione delle quote di frequenza di richiesta API ... 1120
Identificazione dei requisiti delle quote ... 1120
Ottimizza i tassi di richiesta ... 1120

Monitoraggio dell'uso delle quote 1122
Tieni traccia degli utenti attivi mensili (MAU) 1123
Richiesta di aumento delle quote 1123
Quote di frequenza di richiesta di pool di utenti 1124
Quote di frequenza di richiesta di pool di identità 1135
Quote relative al numero e alla dimensione delle risorse 1137
Riferimenti alle API 1144
Documentazione di riferimento degli endpoint del pool di utenti 1144
Documentazione di riferimento degli endpoint dell'interfaccia utente ospitata 1145
Documentazione di riferimento degli endpoint di federazione 1153
Concessioni OAuth 2.0 1178
Usare PKCE 1180
Risposte agli errori di federazione e dell'interfaccia utente ospitata 1182
Documentazione di riferimento dell'API dei bacini d'utenza 1184
Documentazione di riferimento dell'API dei pool di identità 1184
Documentazione di riferimento dell'API di Cognito sync 1184
Cronologia dei documenti 1186
..... mcciii

Che cos'è Amazon Cognito?

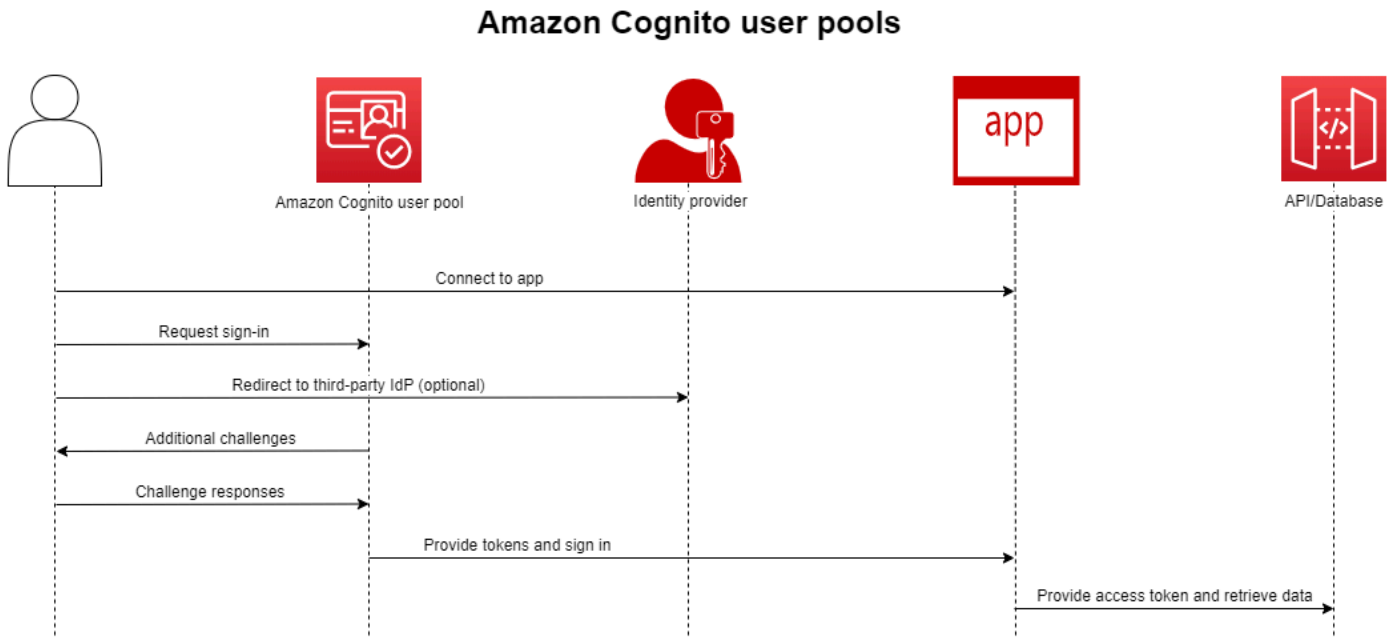
Amazon Cognito è una piattaforma di identità per app web e per dispositivi mobili. È una directory utente, un server di autenticazione e un servizio di autorizzazione per i token di accesso OAuth 2.0 e le credenziali AWS . Con Amazon Cognito, puoi autenticare e autorizzare gli utenti dalla directory utente integrata, dalla directory aziendale e dai provider di identità utente come Google e Facebook.

Argomenti

- Bacini d'utenza
- Pool di identità
- Caratteristiche di Amazon Cognito
- Confronto tra pool di utenti e pool di identità di Amazon Cognito
- Nozioni di base su Amazon Cognito
- Disponibilità regionale
- Prezzi di Amazon Cognito
- Come funziona l'autenticazione con i pool di utenti e i pool di identità di Amazon Cognito
- Termini di Amazon Cognito
- Utilizzo di questo servizio con un SDK AWS
- Iniziare con AWS

I due componenti che seguono costituiscono Amazon Cognito. Funzionano in maniera indipendente o in tandem, in base alle esigenze di accesso degli utenti.

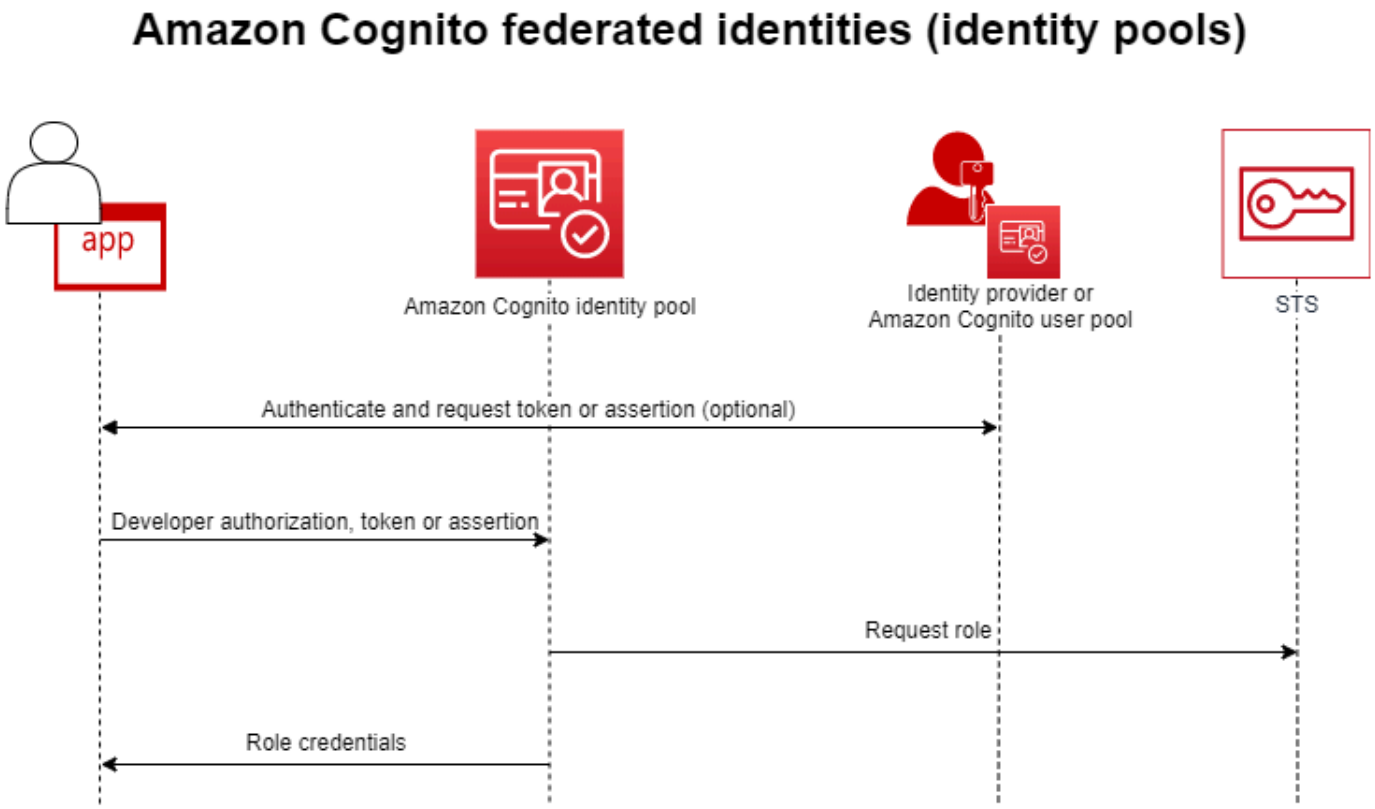
Bacini d'utenza



Crea un pool di utenti quando desideri autenticare e autorizzare gli utenti per l'app o l'API. I pool di utenti sono una directory di utenti con creazione, gestione e autenticazione degli utenti sia in modalità self-service che gestita dagli amministratori. Il pool di utenti può essere una directory indipendente e gestore dell'identità digitale OIDC e un provider di servizi intermedio (SP) per provider di terze parti di identità della forza lavoro e dei clienti. Puoi fornire il Single Sign-On (SSO) nella tua app per le identità della forza lavoro della tua organizzazione in SAML 2.0 e OIDC con pool di utenti. IdPs Puoi anche fornire l'SSO nella tua app per le identità dei clienti dell'organizzazione negli archivi di identità OAuth 2.0 Amazon, Google, Apple e Facebook. Per ulteriori informazioni sulla gestione dell'identità e degli accessi dei clienti (CIAM) consulta Cos'è CIAM?.

I pool di utenti non richiedono l'integrazione con un pool di identità. Da un pool di utenti, puoi emettere token web JSON (JWT) autenticati direttamente in un'app, un server web o un'API.

Pool di identità



Configura un pool di identità Amazon Cognito quando desideri autorizzare utenti autenticati o anonimi ad accedere alle tue risorse. AWS Un pool di identità rilascia AWS le credenziali della tua app per fornire risorse agli utenti. Puoi autenticare gli utenti con un provider di identità attendibile, come un pool di utenti o un servizio SAML 2.0. Facoltativamente, puoi anche emettere credenziali per gli utenti guest. I pool di identità utilizzano il controllo degli accessi basato sui ruoli e sugli attributi per gestire l'autorizzazione degli utenti ad accedere alle risorse. AWS

I pool di utenti non richiedono l'integrazione con un pool di identità. Un pool di identità può accettare richieste autenticate direttamente dai provider di identità della forza lavoro e degli utenti.

Un pool di utenti e un pool di identità di Amazon Cognito utilizzati insieme

Nel diagramma all'inizio di questo argomento, Amazon Cognito viene utilizzato per autenticare l'utente e quindi concedere l'accesso a un Servizio AWS.

1. L'utente dell'app accede tramite un pool di utenti e riceve token OAuth 2.0.

2. La tua app scambia un token del pool di utenti con un pool di identità per AWS credenziali temporanee che puoi utilizzare con le API e (). AWS AWS Command Line Interface AWS CLI
3. L'app assegna la sessione delle credenziali all'utente e fornisce l'accesso autorizzato ad Servizi AWS Amazon S3 e Amazon DynamoDB.

Per ulteriori esempi che utilizzano pool di identità e pool di utenti, consulta Scenari comuni di Amazon Cognito.

In Amazon Cognito, l'obbligo della sicurezza del cloud del modello di responsabilità condivisa è conforme a SOC 1-3, PCI DSS, ISO 27001 ed è idoneo ai fini HIPAA-BAA. Puoi progettare la sicurezza nel cloud in Amazon Cognito in modo che sia conforme a SOC1-3, ISO 27001 e HIPAA-BAA, ma non PCI DSS. Per ulteriori informazioni, consulta Servizi AWS coperti. Consultare anche Considerazioni sui dati regionali.

Caratteristiche di Amazon Cognito

Bacini d'utenza

Un pool di utenti Amazon Cognito è una directory di utenti. Con un pool di utenti, gli utenti possono accedere all'app web o per dispositivi mobili tramite Amazon Cognito o eseguire la federazione tramite un IdP di terze parti. Gli utenti federati e locali hanno un profilo utente nel pool di utenti.

Gli utenti locali sono quelli che hanno effettuato al registrazione o che sono stati creati direttamente nel pool di utenti. Puoi gestire e personalizzare questi profili utente in AWS Management Console, un AWS SDK o (). AWS Command Line Interface AWS CLI

I pool di utenti di Amazon Cognito accettano token e asserzioni di terze parti IdPs e raccolgono gli attributi utente in un JWT che invia alla tua app. Puoi standardizzare la tua app su un set di JWT mentre Amazon Cognito gestisce le interazioni con IdPs, mappando le relative affermazioni su un formato di token centrale.

Un pool di utenti Amazon Cognito può essere un IdP autonomo. Amazon Cognito attinge dallo standard OpenID Connect (OIDC) per generare JWT per l'autenticazione e l'autorizzazione. Quando accedi agli utenti locali, il pool di utenti è autorevole per tali utenti. Quando esegui l'autenticazione degli utenti locali, hai accesso alle seguenti funzionalità.

- Implementa il tuo front-end web che chiama l'API dei pool di utenti Amazon Cognito per autenticare, autorizzare e gestire gli utenti.

- Configura l'autenticazione a più fattori (MFA) per gli utenti. Amazon Cognito supporta password monouso e l'autenticazione MFA con messaggio SMS.
- Proteggi dall'accesso da account utente controllati da malintenzionati.
- Crea i tuoi flussi di autenticazione a più fasi personalizzati.
- Cerca gli utenti in un'altra directory ed esegui la migrazione ad Amazon Cognito.

Un pool di utenti di Amazon Cognito può anche svolgere un duplice ruolo di fornitore di servizi (SP) per la tua IdPs app e di IdP per la tua app. I pool di utenti di Amazon Cognito possono connettersi a consumatori IdPs come Facebook e Google o a forza lavoro IdPs come Okta e Active Directory Federation Services (ADFS).

Con i token OAuth 2.0 e OpenID Connect (OIDC) emessi da un pool di utenti Amazon Cognito, puoi

- Accettare un ID token nell'app che autentica un utente e fornisce le informazioni necessarie per configurare il profilo dell'utente
- Accettare un token di accesso nell'API con gli ambiti OIDC che autorizzano le chiamate API degli utenti.
- Recupera AWS le credenziali da un pool di identità di Amazon Cognito.

Funzionalità del pool di utenti Amazon Cognito

Funzionalità	Descrizione
IdP OIDC	Emetti token ID per autenticare gli utenti
Server di autorizzazione	Emetti token di accesso per autorizzare l'accesso degli utenti alle API
SAML 2.0 SP	Trasforma le asserzioni SAML in ID e token di accesso
OIDC SP	Trasforma i token OIDC in token ID e di accesso
OAuth 2.0 SP	Trasforma i token ID di Apple, Facebook, Amazon o Google nel tuo ID e accedi ai token

Servizio frontend di autenticazione	Registra, gestisci e autentica gli utenti con l'interfaccia utente ospitata
Supporto API per la tua interfaccia utente	Crea, gestisci e autentica gli utenti tramite richieste API negli SDK supportati[1] AWS
MFA	Usa messaggi SMS, TOTP o il dispositivo dell'utente come fattore di autenticazione aggiunto[1]
Monitoraggio e risposta della sicurezza	Proteggiti da attività dannose e password non sicure[1]
Personalizza i flussi di autenticazione	Crea il tuo meccanismo di autenticazione o aggiungi passaggi personalizzati ai flussi esistenti[1]
Gruppi	Crea raggruppamenti logici di utenti e una gerarchia di richieste di ruolo IAM quando passi i token ai pool di identità
Personalizza i token ID	Personalizza i tuoi token ID con attestazioni nuove, modificate e soppresse
Personalizza gli attributi utente	Assegna valori agli attributi utente e aggiungi i tuoi attributi personalizzati

[1] La funzionalità è disponibile solo per gli utenti locali.

Per ulteriori informazioni sui bacini d'utenza, consulta Nozioni di base sui bacini d'utenza e Documentazione di riferimento delle API dei bacini d'utenza di Amazon Cognito.

Pool di identità

Un pool di identità è una raccolta di identificatori univoci, o identità, che assegni ai tuoi utenti o ospiti e autorizzi a ricevere credenziali temporanee. AWS Quando presenti una prova di autenticazione a un pool di identità sotto forma di richieste attendibili da un gestore dell'identità digitale social SAML 2.0, OpenID Connect (OIDC) o OAuth 2.0, associ l'utente a un'identità nel pool di identità. Il token

creato dal pool di identità per l'identità può recuperare le credenziali di sessione temporanee da (). AWS Security Token Service AWS STS

A complemento delle identità autenticate, puoi anche configurare un pool di identità per autorizzare l'accesso AWS senza autenticazione IdP. Puoi offrire una prova di autenticazione personalizzata o nessuna autenticazione. Puoi concedere AWS credenziali temporanee a qualsiasi utente dell'app che le richieda, con identità non autenticate. I pool di identità accettano anche richieste ed emettono credenziali in base a uno schema personalizzato, con identità autenticate dagli sviluppatori.

Con i pool di identità di Amazon Cognito, hai due modi per eseguire l'integrazione con le policy IAM nel Account AWS. Puoi utilizzare queste due funzionalità insieme o singolarmente.

Controllo degli accessi basato sui ruoli

Quando un utente passa le richieste al pool di identità, Amazon Cognito sceglie il ruolo IAM richiesto. Per personalizzare le autorizzazioni del ruolo in base alle esigenze, le policy IAM vengono applicate a ciascun ruolo. Ad esempio, se un utente dimostra di lavorare nel reparto marketing, riceve le credenziali per un ruolo con policy personalizzate in base alle esigenze di accesso del reparto marketing. Amazon Cognito può richiedere un ruolo predefinito, un ruolo basato su regole che eseguono query delle richieste dell'utente o un ruolo basato sull'appartenenza al gruppo dell'utente in un pool di utenti. Puoi anche configurare la policy di attendibilità del ruolo in modo che IAM consideri attendibile solo il tuo pool di identità per generare sessioni temporanee.

Attributi per il controllo degli accessi

Il pool di identità legge gli attributi dalle richieste dell'utente e li associa ai tag principali nella sessione temporanea dell'utente. Puoi quindi configurare le policy basate sulle risorse IAM per consentire o negare l'accesso alle risorse in base ai principali IAM che contengono i tag di sessione del pool di identità. Ad esempio, se l'utente dimostra di lavorare nel reparto marketing, tagga la sua sessione. AWS STS `Department: marketing` Il bucket Amazon S3 consente operazioni di lettura basate su una PrincipalTag condizione aws: che richiede un valore di `marketing` per il tag. `Department`

Funzionalità del pool di identità di Amazon Cognito

Funzionalità	Descrizione
Pool di utenti Amazon Cognito SP	Scambia un token ID dal tuo pool di utenti con credenziali di identità web di AWS STS

SAML 2.0 SP	Scambia asserzioni SAML con credenziali di identità web da AWS STS
OIDC SP	Scambia token OIDC con credenziali di identità web da AWS STS
OAuth 2.0 SP	Scambia i token OAuth di Amazon, Facebook, Google, Apple e Twitter con credenziali di identità web da AWS STS
SP personalizzato	Con AWS le credenziali, puoi scambiare attestazioni in qualsiasi formato con credenziali di identità web da AWS STS
Accesso non autenticato	Emetti credenziali di identità web ad accesso limitato senza autenticazione AWS STS
Controllo degli accessi basato sui ruoli	Scegli un ruolo IAM per il tuo utente autenticato in base alle sue dichiarazioni e configura i ruoli in modo che vengano assunti solo nel contesto del tuo pool di identità
Controllo dell'accesso basato sugli attributi	Converti le attestazioni in tag principali per la tua sessione AWS STS temporanea e utilizza le policy IAM per filtrare l'accesso alle risorse in base ai tag principali

Per ulteriori informazioni sui pool di identità, consulta Guida introduttiva ai pool di identità di Amazon Cognito e Documentazione di riferimento delle API dei pool di identità di Amazon Cognito.

Confronto tra pool di utenti e pool di identità di Amazon Cognito

Funzionalità	Descrizione	Bacini d'utenza	Pool di identità

IdP OIDC	Emetti token ID OIDC per autenticare gli utenti dell'app	✓	
Server di autorizzazione API	Emetti token di accesso per autorizzare l'accesso degli utenti ad API, database e altre risorse che accettano gli ambiti di autorizzazione OAuth 2.0	✓	
Server di autorizzazione dell'identità Web IAM	Genera token con AWS STS cui puoi scambiare credenziali temporanee AWS		✓
SP SAML 2.0 e IdP OIDC	Emetti token OIDC personalizzati in base alle dichiarazioni di un IdP SAML 2.0	✓	
OIDC SP e OIDC IdP	Emetti token OIDC personalizzati in base alle affermazioni di un IdP OIDC	✓	
OAuth 2.0 SP e IdP OIDC	Emetti token OIDC personalizzati in base agli ambiti dei social provider OAuth 2.0 come Apple e Google	✓	
SP SAML 2.0 e broker di credenziali	Emetti AWS credenziali temporanee basate sulle attestazioni di un IdP SAML 2.0		✓

Broker di credenziali e SP OIDC	Emetti AWS credenziali temporanee basate sulle dichiarazioni di un IdP OIDC		✓
SP e broker di credenziali OAuth 2.0	Emetti AWS credenziali temporanee basate sugli ambiti dei social provider OAuth 2.0 come Apple e Google		✓
SP e broker di credenziali per pool di utenti Amazon Cognito	Emetti AWS credenziali temporanee basate sulle dichiarazioni OIDC di un pool di utenti di Amazon Cognito		✓
Broker di credenziali e SP personalizzato	Emetti AWS credenziali temporanee basate sull'autorizzazione IAM dello sviluppatore		✓
Servizio frontend di autenticazione	Registra, gestisci e autentica gli utenti con l'interfaccia utente ospitata	✓	
Supporto API per la tua interfaccia utente di autenticazione	Crea, gestisci e autentica gli utenti tramite richieste API negli SDK supportati[1] AWS	✓	

MFA	Usa messaggi SMS, TOTP o il dispositivo dell'utente come fattore di autenticazione aggiunto[1]	✓
Monitoraggio e risposta della sicurezza	Proteggiti da attività dannose e password non sicure[1]	✓
Personalizza i flussi di autenticazione	Crea il tuo meccanismo di autenticazione o aggiungi passaggi personalizzati ai flussi esistenti[1]	✓
Gruppi	Crea raggruppamenti logici di utenti e una gerarchia di richieste di ruolo IAM quando passi i token ai pool di identità	✓
Personalizza i token ID	Personalizza i tuoi token ID con attestazioni nuove, modificate e soppresse	✓
AWS WAF ACL web	Monitora e controlla le richieste al tuo ambiente di autenticazione con AWS WAF	✓
Personalizza gli attributi utente	Assegna valori agli attributi utente e aggiungi i tuoi attributi personalizzati	✓

Accesso non autenticato	Emetti credenziali di identità web ad accesso limitato senza autenticazione AWS STS	✓
Controllo degli accessi basato sui ruoli	Scegli un ruolo IAM per il tuo utente autenticato in base alle sue dichiaraz ioni e configura i ruoli in modo che vengano assunti solo nel contesto del tuo pool di identità	✓
Controllo dell'accesso basato sugli attributi	Trasforma le affermazi oni degli utenti in tag principali per la sessione AWS STS temporanea e utilizza le policy IAM per filtrare l'accesso alle risorse in base ai tag principali	✓

[1] La funzionalità è disponibile solo per gli utenti locali.

Nozioni di base su Amazon Cognito

Ad esempio, applicazioni con pool di utenti, vediNozioni di base sui bacini d'utenza.

Per un'introduzione ai pool di identità, vedereGuida introduttiva ai pool di identità di Amazon Cognito.

Per i collegamenti alle esperienze di configurazione guidate con pool di utenti e pool di identità, consultaOpzioni di configurazione guidate per Amazon Cognito.

Per video, articoli, documentazione e altre applicazioni di esempio, consulta le risorse per sviluppatori di Amazon Cognito.

Per utilizzare Amazon Cognito devi disporre di un Account AWS. Per ulteriori informazioni, consulta Iniziare con AWS.

Disponibilità regionale

Amazon Cognito è disponibile in diverse AWS regioni in tutto il mondo. In ciascuna regione, Amazon Cognito viene distribuito su più zone di disponibilità. Queste zone di disponibilità sono fisicamente isolate l'una dall'altra, ma sono unite da connessioni di rete private a bassa latenza, a velocità effettiva elevata e altamente ridondanti. Queste zone di disponibilità consentono AWS di fornire servizi, tra cui Amazon Cognito, con livelli molto elevati di disponibilità e ridondanza, riducendo al contempo al minimo la latenza.

Per l'elenco di tutte le regioni nelle quali Amazon Cognito è disponibile, vedi Regioni ed endpoint AWS nella Riferimenti generali di Amazon Web Services. Per ulteriori informazioni sul numero di zone di disponibilità presenti in ciascuna regione, consulta Infrastruttura globale AWS.

Prezzi di Amazon Cognito

Per informazioni sui prezzi di Amazon Cognito, consulta Prezzi di Amazon Cognito.

Come funziona l'autenticazione con i pool di utenti e i pool di identità di Amazon Cognito

Quando il cliente accede a un pool di utenti Amazon Cognito, l'applicazione riceve token web JSON (JWT).

Quando il cliente accede a un pool di identità, con un token del pool di utenti o con un altro provider, l'applicazione riceve credenziali temporanee. AWS

Con l'accesso al pool di utenti, puoi implementare l'autenticazione e l'autorizzazione interamente con un AWS SDK. Se non desideri creare componenti di interfaccia utente (UI) personalizzati, puoi richiamare un'interfaccia utente Web predefinita (l'interfaccia utente ospitata) o la pagina di accesso per il tuo provider di identità (IdP) di terze parti.

Questo argomento offre una panoramica di alcuni modi in cui l'applicazione può interagire con Amazon Cognito per autenticarsi con token ID, autorizzare con token di accesso e accedere con credenziali di pool di identità. Servizi AWS

Argomenti

- Autenticazione e autorizzazione dell'API del pool di utenti con un SDK AWS
- Autenticazione del pool di utenti con l'interfaccia utente ospitata
- Autenticazione del pool di utenti con un provider di identità di terze parti
- Autenticazione del pool di identità

Autenticazione e autorizzazione dell'API del pool di utenti con un SDK AWS

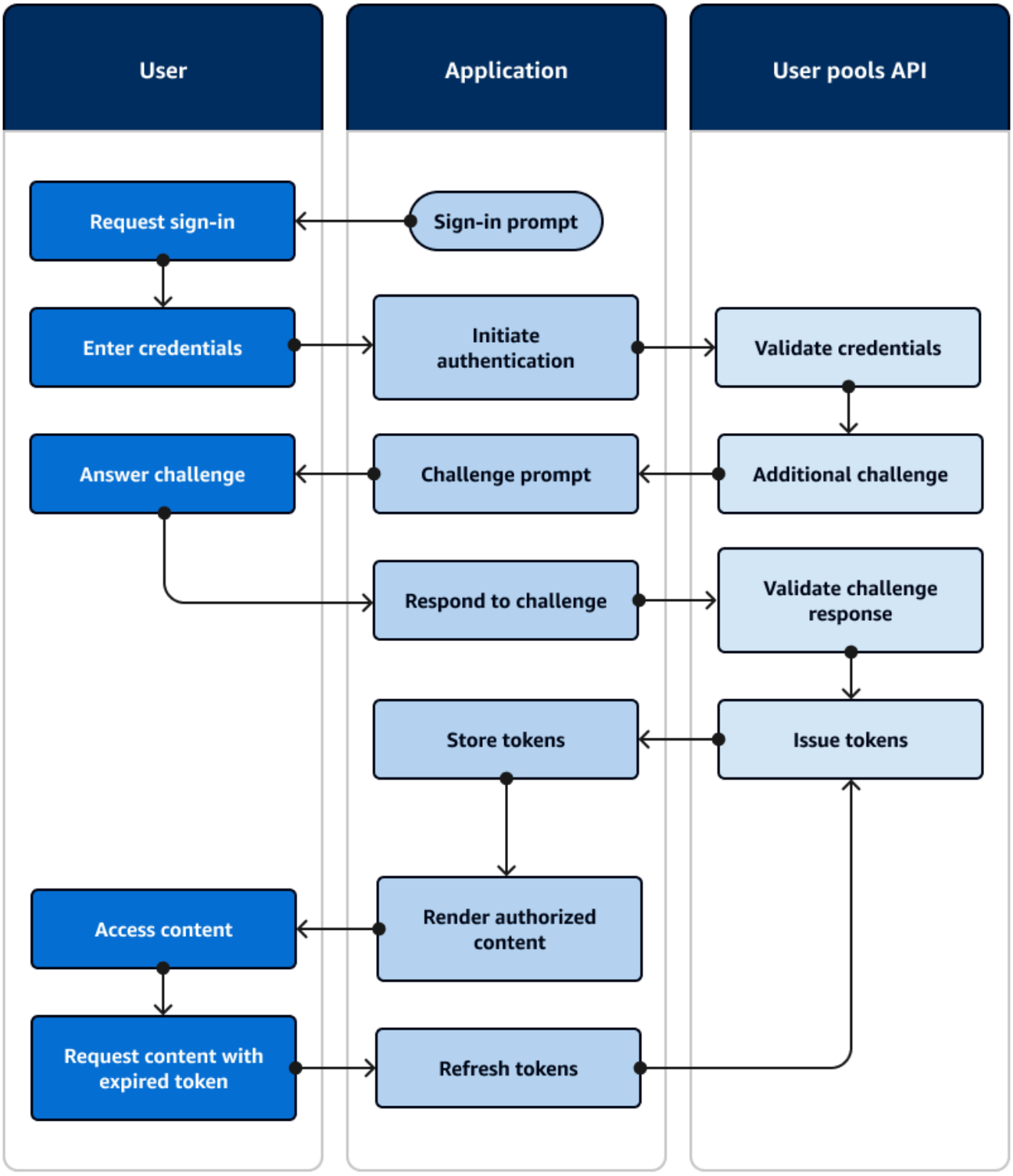
AWS ha sviluppato componenti per i pool di utenti di Amazon Cognito, o provider di identità Amazon Cognito, in una varietà di framework di sviluppo. I metodi integrati in questi SDK chiamano l'API dei pool di utenti di Amazon Cognito. Lo stesso spazio dei nomi API dei pool di utenti include operazioni per la configurazione dei pool di utenti e per l'autenticazione degli utenti. Per una panoramica più completa, vedere. Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti

L'autenticazione API si adatta al modello in cui le applicazioni dispongono di componenti dell'interfaccia utente esistenti e si basano principalmente sul pool di utenti come directory utente. Questo design aggiunge Amazon Cognito come componente all'interno di un'applicazione più grande. Richiede una logica programmatica per gestire catene complesse di sfide e risposte.

Questa applicazione non ha bisogno di implementare un'implementazione completa del relying party OpenID Connect (OIDC). Invece, ha la capacità di decodificare e utilizzare JWT. Se desideri accedere al set completo di funzionalità del pool di utenti per gli utenti locali, crea la tua autenticazione con l'SDK Amazon Cognito nel tuo ambiente di sviluppo.

L'autenticazione API con ambiti OAuth personalizzati è meno orientata all'autorizzazione API esterna. Per aggiungere ambiti personalizzati a un token di accesso dall'autenticazione API, modifica il token in fase di esecuzione con un. Trigger Lambda di pre-generazione del token

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione tramite API.


User
Application
User pools API
Request sign-in
Sign-in prompt
Enter credentials
Initiate authentication
Validate credentials
Answer challenge
Challenge prompt
Additional challenge
Respond to challenge
Validate challenge response
Store tokens
Issue tokens
Access content
Render authorized content
Request content with expired token
Refresh tokens

Flusso di autenticazione dell'API

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».
3. Inseriscono nome utente e password.
4. L'applicazione richiama il metodo che effettua una richiesta InitiateAuthAPI. La richiesta passa le credenziali dell'utente a un pool di utenti.
5. Il pool di utenti convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. Il pool di utenti risponde con una sfida che richiede un codice MFA.
7. L'applicazione genera un prompt che raccoglie il codice MFA dall'utente.
8. L'applicazione richiama il metodo che effettua una richiesta API. RespondToAuthChallenge La richiesta passa il codice MFA dell'utente.
9. Il pool di utenti convalida il codice MFA dell'utente.
10. Il pool di utenti risponde con i JWT dell'utente.
11. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
12. L'applicazione visualizza il componente con accesso controllato richiesto.
13. L'utente ne visualizza il contenuto.
14. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
15. L'applicazione determina che la sessione dell'utente debba persistere. Richiama nuovamente il InitiateAuthmetodo con il token di aggiornamento e recupera nuovi token.

Varianti e personalizzazioni

Puoi ampliare questo flusso con sfide aggiuntive, ad esempio sfide di autenticazione personalizzate. Puoi limitare automaticamente l'accesso agli utenti le cui password sono state compromesse o le cui caratteristiche di accesso impreviste potrebbero indicare un tentativo di accesso malevolo. Questo flusso ha lo stesso aspetto per le operazioni di registrazione, aggiornamento degli attributi utente e reimpostazione delle password. La maggior parte di questi flussi prevede operazioni API pubbliche (lato client) e riservate (lato server) duplicate.

Risorse correlate

- API dei pool di utenti di Amazon Cognito

- Nozioni di base sui bacini d'utenza
- Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili
- Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti

Autenticazione del pool di utenti con l'interfaccia utente ospitata

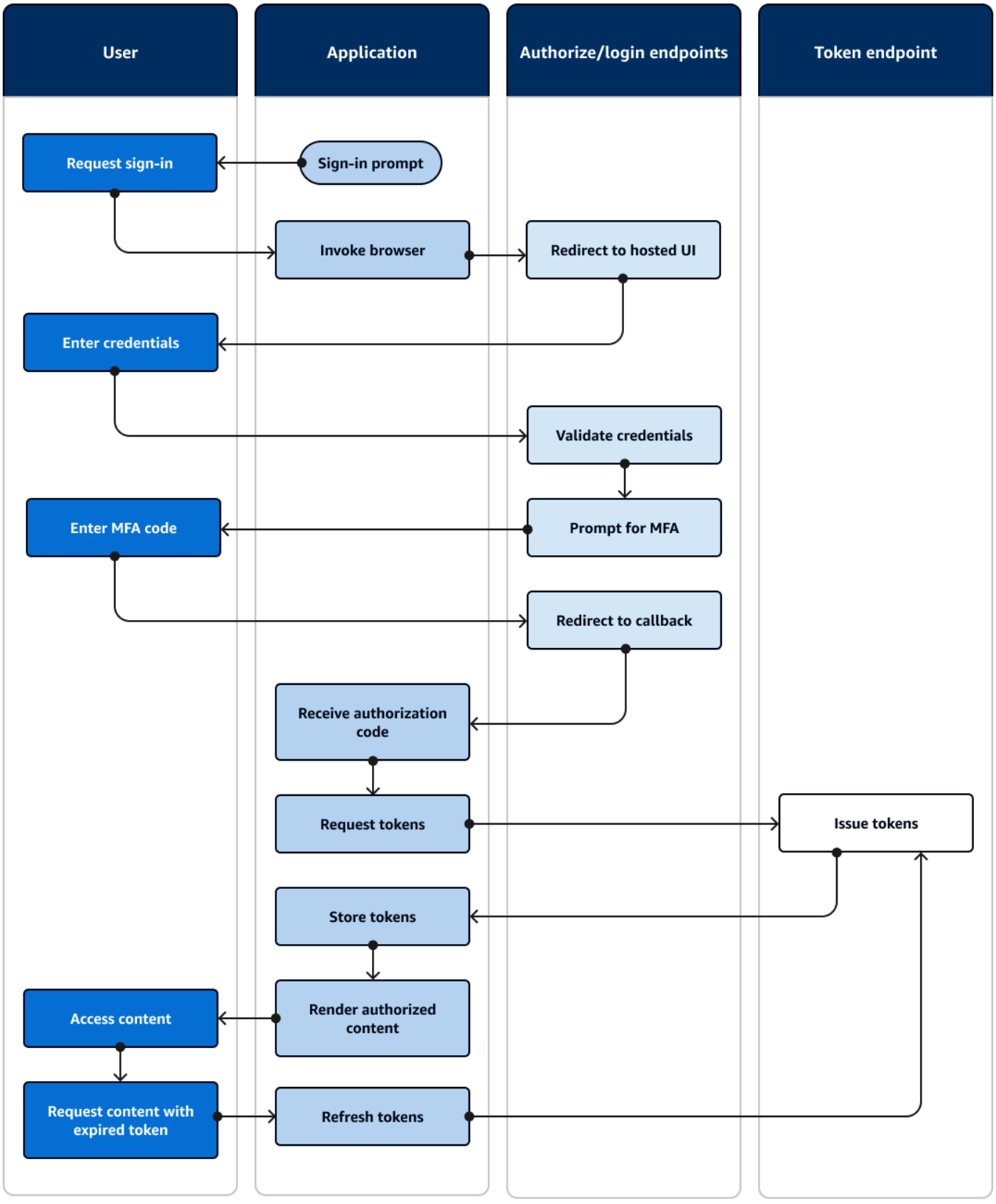
L'interfaccia utente ospitata è un sito Web collegato al pool di utenti e al client dell'app. Può eseguire operazioni di accesso, registrazione e reimpostazione della password per i tuoi utenti. Un'applicazione con un componente dell'interfaccia utente ospitato per l'autenticazione può richiedere meno sforzi da parte degli sviluppatori per l'implementazione. Un'applicazione può ignorare i componenti dell'interfaccia utente per l'autenticazione e richiamare l'interfaccia utente ospitata nel browser dell'utente.

Le applicazioni raccolgono i JWT degli utenti con una posizione di reindirizzamento Web o app. Le applicazioni che implementano l'interfaccia utente ospitata possono connettersi ai pool di utenti per l'autenticazione come se fossero un IdP OpenID Connect (OIDC).

L'autenticazione dell'interfaccia utente ospitata si adatta al modello in cui le applicazioni necessitano di un server di autorizzazione, ma non necessitano di funzionalità come l'autenticazione personalizzata, l'integrazione dei pool di identità o il self-service per gli attributi utente. Se desideri utilizzare alcune di queste opzioni avanzate, puoi implementarle con un componente di pool di utenti per un SDK.

L'interfaccia utente ospitata e i modelli di autenticazione IdP di terze parti, che si basano principalmente sull'implementazione OIDC, sono ideali per i modelli di autorizzazione avanzati con ambiti OAuth 2.0.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione dell'interfaccia utente ospitata.


User
Application
Authorize/login endpoints
Token endpoint
Request sign-in
Sign-in prompt
Invoke browser
Redirect to hosted UI
Enter credentials
Validate credentials
Enter MFA code
Prompt for MFA
Redirect to callback
Receive authorization code
Request tokens
Issue tokens
Store tokens
Access content
Render authorized content
Request content with expired token
Refresh tokens

Flusso di autenticazione dell'interfaccia utente ospitata

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».
3. L'applicazione indirizza l'utente a una richiesta di accesso all'interfaccia utente ospitata.
4. Inseriscono nome utente e password.
5. Il pool di utenti convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. L'interfaccia utente ospitata richiede all'utente di inserire un codice MFA.
7. L'utente inserisce il proprio codice MFA.
8. L'interfaccia utente ospitata reindirizza l'utente all'applicazione.
9. L'applicazione raccoglie il codice di autorizzazione dal parametro di richiesta URL che l'interfaccia utente ospitata ha aggiunto all'URL di callback.
10. L'applicazione richiede i token con il codice di autorizzazione.
11. L'endpoint del token restituisce JWT all'applicazione.
12. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
13. L'applicazione visualizza il componente con accesso controllato richiesto.
14. L'utente ne visualizza il contenuto.
15. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
16. L'applicazione determina che la sessione dell'utente debba persistere. Richiede nuovi token dall'endpoint del token con il token di aggiornamento.

Varianti e personalizzazioni

Puoi personalizzare l'aspetto dell'interfaccia utente ospitata con CSS in qualsiasi client dell'app. Puoi anche configurare i client delle app con i propri provider di identità, ambiti, accesso agli attributi utente e configurazioni di sicurezza avanzate.

Risorse correlate

- Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione
- Registrazione e accesso con l'interfaccia utente ospitata

- Autorizzazione Scopes, M2M e API con server di risorse
- Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata

Autenticazione del pool di utenti con un provider di identità di terze parti

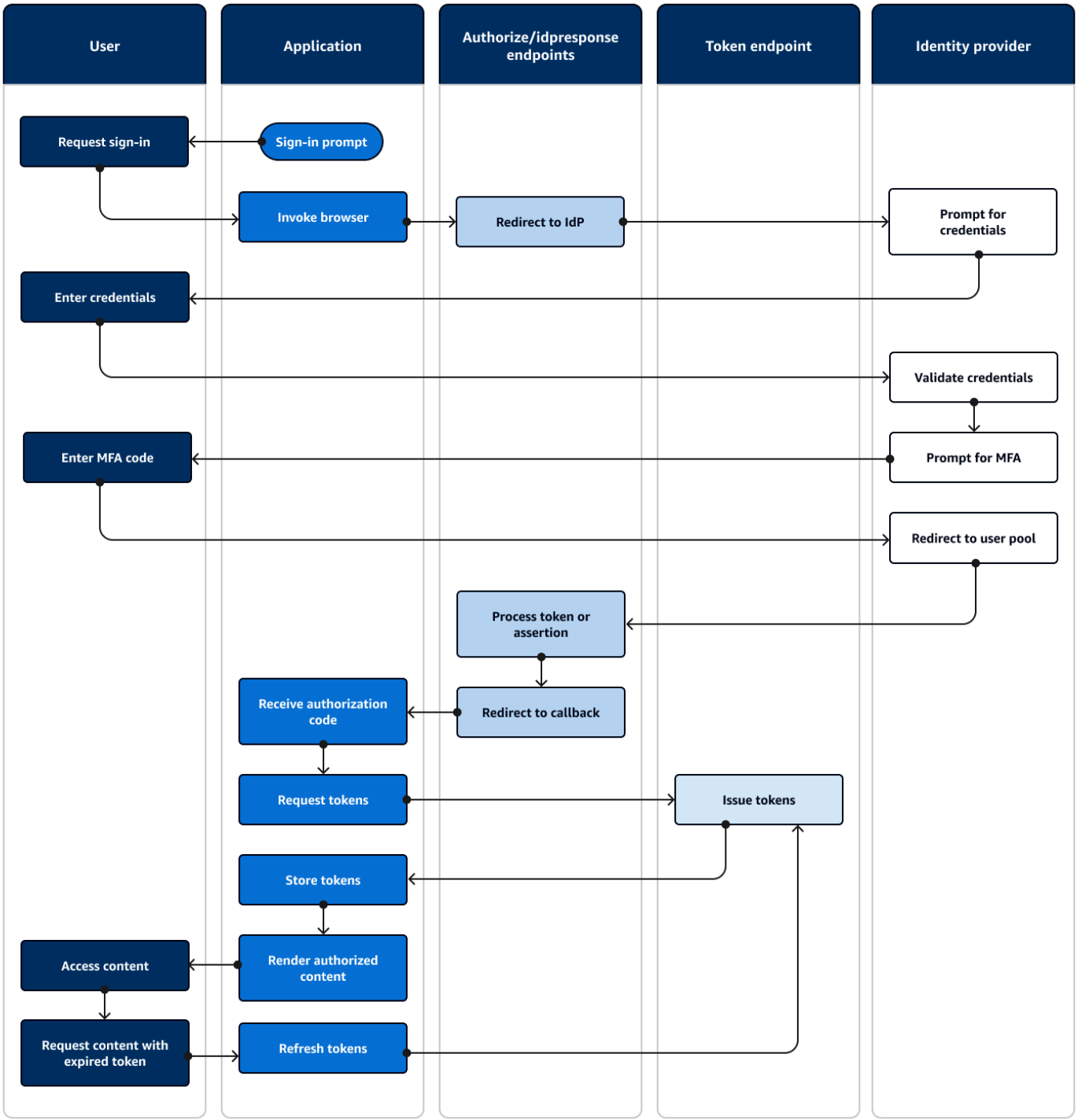
L'accesso con un provider di identità esterno (IdP), o autenticazione federata, è un modello simile all'interfaccia utente ospitata. La tua applicazione è un relying party OIDC per il tuo pool di utenti, mentre il pool di utenti funge da passthrough per un IdP. L'IdP può essere una directory di utenti consumer come Facebook o Google, oppure può essere una directory aziendale SAML 2.0 o OIDC come Azure.

Invece dell'interfaccia utente ospitata nel browser dell'utente, l'applicazione richiama un endpoint di reindirizzamento sul server di autorizzazione del pool di utenti. Dal punto di vista dell'utente, sceglie il pulsante di accesso nell'applicazione. Quindi il loro IdP li richiede di accedere. Come con l'autenticazione dell'interfaccia utente ospitata, un'applicazione raccoglie i JWT in una posizione di reindirizzamento all'interno dell'app.

L'autenticazione con un IdP di terze parti si adatta a un modello in cui gli utenti potrebbero non voler inserire una nuova password quando si iscrivono alla tua applicazione. L'autenticazione di terze parti può essere aggiunta con il minimo sforzo a un'applicazione che implementa l'autenticazione dell'interfaccia utente ospitata. In effetti, l'interfaccia utente ospitata e quella di terze parti IdPs producono un risultato di autenticazione coerente a partire da piccole variazioni di ciò che viene richiamato nei browser degli utenti.

Come l'autenticazione dell'interfaccia utente ospitata, l'autenticazione federata è ideale per i modelli di autorizzazione avanzati con ambiti OAuth 2.0.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione federata.



Flusso di autenticazione federato

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».

3. L'applicazione indirizza l'utente a una richiesta di accesso con il proprio IdP.
4. Inseriscono nome utente e password.
5. L'IdP convalida le credenziali dell'utente e determina che l'utente ha attivato l'autenticazione a più fattori (MFA).
6. L'IdP richiede all'utente di inserire un codice MFA.
7. L'utente inserisce il proprio codice MFA.
8. L'IdP reindirizza l'utente al pool di utenti con una risposta SAML o un codice di autorizzazione.
9. Se l'utente ha passato un codice di autorizzazione, il pool di utenti scambia silenziosamente il codice con token IdP. Il pool di utenti convalida i token IdP e reindirizza l'utente all'applicazione con un nuovo codice di autorizzazione.
10. L'applicazione raccoglie il codice di autorizzazione dal parametro di richiesta URL che il pool di utenti ha aggiunto all'URL di callback.
11. L'applicazione richiede i token con il codice di autorizzazione.
12. L'endpoint del token restituisce JWT all'applicazione.
13. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT dell'utente.
14. L'applicazione visualizza il componente con accesso controllato richiesto.
15. L'utente ne visualizza il contenuto.
16. Successivamente, il token di accesso dell'utente è scaduto e richiede di visualizzare un componente con accesso controllato.
17. L'applicazione determina che la sessione dell'utente debba persistere. Richiede nuovi token dall'endpoint del token con il token di aggiornamento.

Varianti e personalizzazioni

Puoi avviare l'autenticazione federata nell'interfaccia utente ospitata, dove gli utenti possono scegliere da un elenco di quelle IdPs che hai assegnato al client dell'app. L'interfaccia utente ospitata può anche richiedere un indirizzo e-mail e indirizzare automaticamente la richiesta di un utente all'IdP SAML corrispondente. L'autenticazione con un provider di identità di terze parti non richiede l'interazione dell'utente con l'interfaccia utente ospitata. L'applicazione può aggiungere un parametro di richiesta alla richiesta del server di autorizzazione dell'utente e fare in modo che l'utente reindirizzi silenziosamente alla pagina di accesso dell'IdP.

Risorse correlate

- Aggiunta di un accesso al bacino d'utenza tramite terze parti

- Scenario di esempio: aggiungi ai preferiti le app Amazon Cognito in una dashboard aziendale
- Autorizzazione Scopes, M2M e API con server di risorse
- Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata

Autenticazione del pool di identità

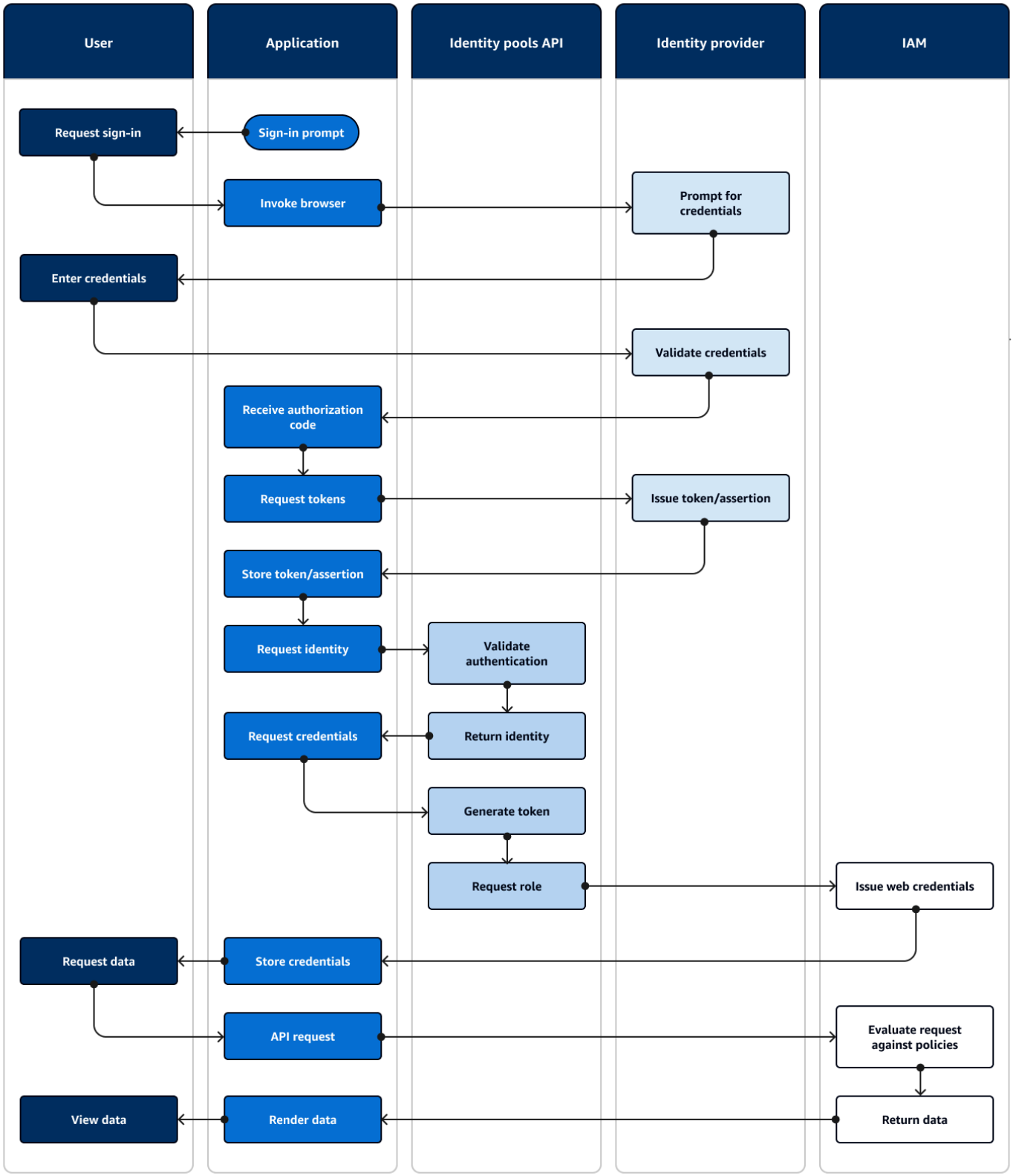
Un pool di identità è un componente dell'applicazione che si distingue da un pool di utenti in termini di funzioni, namespace API e modello SDK. Laddove i pool di utenti offrono l'autenticazione e l'autorizzazione basate su token, i pool di identità offrono l'autorizzazione per (IAM). AWS Identity and Access Management

Puoi assegnarne un set IdPs ai pool di identità e accedere agli utenti con essi. I pool di utenti sono strettamente integrati come pool di identità IdPs e offrono ai pool di identità la maggior parte delle opzioni per il controllo degli accessi. Allo stesso tempo, esiste un'ampia selezione di opzioni di autenticazione per i pool di identità. I pool di utenti si uniscono alle fonti di identità SAML, OIDC, social, per sviluppatori e ospiti come percorsi verso le AWS credenziali temporanee dai pool di identità.

L'autenticazione con un pool di identità è esterna: segue uno dei flussi del pool di utenti illustrati in precedenza o un flusso sviluppato indipendentemente con un altro IdP. Dopo aver eseguito l'autenticazione iniziale, l'applicazione passa la prova a un pool di identità e riceve in cambio una sessione temporanea.

L'autenticazione con un pool di identità si adatta a un modello in cui si impone il controllo degli accessi per gli asset e i dati delle applicazioni Servizi AWS con l'autorizzazione IAM. Analogamente all'autenticazione tramite API nei pool di utenti, un'applicazione di successo include AWS SDK per ciascuno dei servizi a cui desideri accedere a vantaggio degli utenti. AWS Gli SDK applicano le credenziali dell'autenticazione del pool di identità come firme alle richieste API.

Il diagramma seguente illustra una tipica sessione di accesso per l'autenticazione del pool di identità con un IdP.


User
Application
Identity pools API
Identity provider
IAM
Request sign-in
Sign-in prompt
Invoke browser
Prompt for credentials
Enter credentials
Validate credentials
Receive authorization code
Request tokens
Issue token/assertion
Store token/assertion
Request identity
Validate authentication
Request credentials
Return identity
Generate token
Request role
Issue web credentials
Request data
Store credentials
API request
Evaluate request against policies
View data
Render data
Return data

Flusso di autenticazione federato

1. Un utente accede alla tua applicazione.
2. Selezionano un link «Accedi».
3. L'applicazione indirizza l'utente a una richiesta di accesso con il proprio IdP.
4. Inseriscono nome utente e password.
5. L'IdP convalida le credenziali dell'utente.
6. L'IdP reindirizza l'utente all'applicazione con una risposta SAML o un codice di autorizzazione.
7. Se l'utente ha passato un codice di autorizzazione, l'applicazione scambia il codice con token IdP.
8. L'applicazione decodifica, convalida e archivia o memorizza nella cache i JWT o l'asserzione dell'utente.
9. L'applicazione richiama il metodo che effettua una richiesta API. GetId Passa il token o l'asserzione dell'utente e richiede un ID di identità.
10. Il pool di identità convalida il token o l'asserzione rispetto ai provider di identità configurati.
11. Il pool di identità restituisce un ID di identità.
12. L'applicazione richiama il metodo che effettua una richiesta GetCredentialsForIdentityAPI. Passa il token o le asserzioni dell'utente e richiede un ruolo IAM.
13. Il pool di identità genera un nuovo JWT. Il nuovo JWT contiene affermazioni che richiedono un ruolo IAM. Il pool di identità determina il ruolo in base alla richiesta dell'utente e ai criteri di selezione del ruolo nella configurazione del pool di identità per l'IdP.
14. AWS Security Token Service (AWS STS) risponde alla AssumeRoleWithWebIdentityrichiesta del pool di identità. La risposta contiene le credenziali API per una sessione temporanea con un ruolo IAM.
15. L'applicazione memorizza le credenziali della sessione.
16. L'utente esegue un'azione nell'app che richiede risorse con accesso protetto in. AWS
17. L'applicazione applica le credenziali temporanee come firme alle richieste API per quanto richiesto. Servizi AWS
18. IAM valuta le politiche associate al ruolo nelle credenziali. Le confronta con la richiesta.
19. Servizio AWS Restituisce i dati richiesti.
20. L'applicazione esegue il rendering dei dati nell'interfaccia utente.
21. L'utente visualizza i dati.

Varianti e personalizzazioni

Per visualizzare l'autenticazione con un pool di utenti, inserisci una delle precedenti panoramiche del pool di utenti dopo la fase Issue token/assertion. L'autenticazione dello sviluppatore sostituisce tutti i passaggi precedenti a Request identity con una richiesta firmata dalle credenziali dello sviluppatore. L'autenticazione guest passa inoltre direttamente a Request identity, non convalida l'autenticazione e restituisce le credenziali per un ruolo IAM ad accesso limitato.

Risorse correlate

- Pool di identità di Amazon Cognito
- Ruoli IAM dell'utente
- Concetti del pool di identità
- Flusso di autenticazione dei pool di identità (identità federate)

Termini di Amazon Cognito

Amazon Cognito fornisce credenziali per app Web e mobili. Attinge e si basa su termini comuni nella gestione delle identità e degli accessi. Sono disponibili molte guide all'identità universale e ai termini di accesso. Alcuni esempi sono:

- Terminologia nel Body of Knowledge di IDPro
- AWS Servizi di identità
- Glossario del NIST CSRC

Gli elenchi seguenti descrivono termini che sono esclusivi di Amazon Cognito o che hanno un contesto specifico in Amazon Cognito.

Argomenti

- Generali
- Bacini d'utenza
- Pool di identità

Generali

I termini in questo elenco non sono specifici di Amazon Cognito e sono ampiamente riconosciuti tra i professionisti della gestione delle identità e degli accessi. Quello che segue non è un elenco esaustivo di termini, ma una guida al contesto specifico di Amazon Cognito contenuto in questa guida.

App

In genere, un'applicazione mobile. In questa guida, app è spesso un'abbreviazione di un'applicazione Web o di un'app mobile che si connette ad Amazon Cognito.

Controllo degli accessi basato su attributi (ABAC)

Un modello in cui un'app determina l'accesso alle risorse in base alle proprietà di un utente, come la qualifica o il reparto. Gli strumenti di Amazon Cognito per applicare l'ABAC includono i token ID nei pool di utenti e i tag principali nei pool di identità.

Server di autorizzazione

Un sistema basato sul Web che genera token web JSON. Gli endpoint federativi dei pool di utenti di Amazon Cognito sono il componente server di autorizzazione dei due metodi di autenticazione e autorizzazione nei pool di utenti. L'altro metodo è l'API dei pool di utenti.

App riservata, app lato server

Un'applicazione a cui gli utenti si connettono in remoto, con codice su un server di applicazioni e accesso ai segreti. Si tratta in genere di un'applicazione Web.

Identity provider (IdP) (Provider di identità)

Un servizio che archivia e verifica le identità degli utenti. Amazon Cognito può richiedere l'autenticazione a provider esterni ed essere un IdP delle app.

Token web JSON (JWT)

Un documento in formato JSON che contiene affermazioni relative a un utente autenticato. I token ID autenticano gli utenti, i token di accesso autorizzano gli utenti e i token di aggiornamento aggiornano le credenziali. Amazon Cognito riceve token da fornitori esterni ed emette token per app o. AWS STS

Autenticazione a più fattori (MFA)

Il requisito che gli utenti forniscano un'autenticazione aggiuntiva dopo aver fornito nome utente e password. I pool di utenti di Amazon Cognito dispongono di funzionalità MFA per gli utenti locali.

Provider OAuth 2.0 (social)

Un IdP verso un pool di utenti o un pool di identità che fornisce token di accesso e aggiornamento JWT. I pool di utenti di Amazon Cognito automatizzano le interazioni con i social provider dopo l'autenticazione degli utenti.

Provider OpenID Connect (OIDC)

Un IdP a un pool di utenti o a un pool di identità che estende la specifica OAuth per fornire token ID. I pool di utenti di Amazon Cognito automatizzano le interazioni con i provider OIDC dopo l'autenticazione degli utenti.

App pubblica

Un'applicazione autonoma su un dispositivo, con codice archiviato localmente e senza accesso ai segreti. Si tratta in genere di un'app per dispositivi mobili.

Server di risorse

Un'API con controllo degli accessi. I pool di utenti di Amazon Cognito utilizzano anche il server di risorse per descrivere il componente che definisce la configurazione per l'interazione con un'API.

Controllo degli accessi basato sui ruoli (RBAC)

Un modello che concede l'accesso in base alla designazione funzionale di un utente. I pool di identità di Amazon Cognito implementano RBAC con differenziazione tra i ruoli IAM.

Fornitore di servizi (SP), relying party (RP)

Un'applicazione che si affida a un IdP per affermare l'affidabilità degli utenti. Amazon Cognito funge da SP per gli SP esterni IdPs e come IdP per gli SP basati su app.

Provider SAML

Un IdP verso un pool di utenti o un pool di identità che genera documenti di asserzione con firma digitale che l'utente trasmette ad Amazon Cognito.

Identificatore univoco universale (UUID)

Un'etichetta a 128 bit applicata a un oggetto. Gli UUID di Amazon Cognito sono unici per pool di utenti o pool di identità.

Elenco utenti

Una raccolta di utenti e relativi attributi che fornisce tali informazioni ad altri sistemi. I pool di utenti di Amazon Cognito sono elenchi di utenti e anche strumenti per il consolidamento degli utenti provenienti da elenchi utenti esterni.

Bacini d'utenza

Quando vedi i termini nell'elenco seguente di questa guida, si riferiscono a una funzionalità o configurazione specifica dei pool di utenti.

API dei pool di utenti di Amazon Cognito

: Una serie di operazioni API di autenticazione e autorizzazione che puoi aggiungere alla tua app con un AWS SDK. L'API può accedere agli utenti locali e agli utenti collegati.

Autenticazione adattiva

: Una funzionalità di sicurezza avanzata che rileva potenziali attività dannose e applica una protezione aggiuntiva ai profili utente.

Funzionalità di sicurezza avanzate

: Un componente opzionale che aggiunge strumenti per la sicurezza degli utenti.

Client dell'app

: Un componente che definisce le impostazioni per un pool di utenti come IdP per un'app.

URL di callback, URI di reindirizzamento

: Un'impostazione in un client di app e un parametro nelle richieste agli endpoint federativi dei pool di utenti. L'URL di callback è la destinazione iniziale per gli utenti autenticati nell'app.

Credenziali compromesse

: Una funzionalità di sicurezza avanzata che rileva le password degli utenti che gli aggressori potrebbero conoscere e applica una protezione aggiuntiva ai profili utente.

Conferma

: Processo che determina che sono stati soddisfatti i prerequisiti per consentire a un nuovo utente di accedere. La conferma viene in genere effettuata tramite la verifica dell'indirizzo e-mail o del numero di telefono.

Autenticazione personalizzata

: Un'estensione dei processi di autenticazione con trigger Lambda che definiscono sfide e risposte aggiuntive per gli utenti.

Autenticazione del dispositivo

: Un processo di autenticazione che sostituisce la MFA con l'accesso che utilizza l'ID di un dispositivo affidabile.

Provider esterno, fornitore terzo

Un IdP che ha una relazione di fiducia con un pool di utenti.

Utente federato

Un utente in un pool di utenti che è stato autenticato da un provider esterno.

Endpoint della federazione

Un set di pagine Web sul dominio del pool di utenti che ospitano servizi per l'interazione con IdPs e app.

Interfaccia utente ospitata

Un set di pagine Web interattive sul dominio del pool di utenti che ospitano servizi per l'autenticazione degli utenti.

Trigger Lambda

Una funzione AWS Lambda che un pool di utenti può richiamare automaticamente nei punti chiave dei processi di autenticazione degli utenti. Puoi utilizzare i trigger Lambda per personalizzare i risultati dell'autenticazione.

Utente locale

Un profilo utente nella directory degli utenti del pool di utenti che non è stato creato mediante l'autenticazione con un provider esterno.

Utente collegato

Un utente di un provider esterno la cui identità viene unita a quella di un utente locale.

Personalizzazione dei token

Il risultato di un trigger Lambda precedente alla generazione del token che modifica l'ID o il token di accesso di un utente in fase di esecuzione.

Pool di utenti, provider di identità Amazon Cognito**`cognito-idp`**, pool di utenti Amazon Cognito

Una AWS risorsa con servizi di autenticazione e autorizzazione per applicazioni che funzionano con OIDC. IdPs

Dominio del pool di utenti

Un nome di sito Web che aggiungi a un pool di utenti. Il dominio è l'URL di base per l'interfaccia utente ospitata e gli endpoint federativi.

Verifica

Il processo di conferma che un utente possiede un indirizzo e-mail o un numero di telefono. Un pool di utenti invia un codice a un utente che ha inserito un nuovo indirizzo e-mail o numero di telefono. Quando inviano il codice ad Amazon Cognito, verificano la proprietà della destinazione del messaggio e possono ricevere messaggi aggiuntivi dal pool di utenti. Inoltre, vedi conferma.

Profilo utente, account utente

Una voce per un utente nella rubrica degli utenti. Tutti gli utenti hanno un profilo nel proprio pool di utenti.

Pool di identità

Quando vedi i termini nell'elenco seguente di questa guida, si riferiscono a una funzionalità o configurazione specifica dei pool di identità.

Attributi per il controllo degli accessi

Un'implementazione del controllo degli accessi basato sugli attributi nei pool di identità. I pool di identità applicano gli attributi utente come tag alle credenziali utente.

Autenticazione di base (classica)

Un processo di autenticazione in cui è possibile personalizzare la richiesta di credenziali utente.

Identità autenticate dagli sviluppatori

Un processo di autenticazione che autorizza le credenziali utente del pool di identità con le credenziali dello sviluppatore.

Credenziali dello sviluppatore

Le chiavi API IAM di un amministratore di pool di identità.

Autenticazione avanzata

Un flusso di autenticazione che seleziona un ruolo IAM e applica i tag principali in base alla logica definita nel pool di identità.

Identità

Un UUID che collega un utente dell'app e le relative credenziali utente al relativo profilo in una directory utente esterna che ha una relazione di fiducia con un pool di identità.

Pool di identità, identità federate Amazon Cognito, identità Amazon Cognito, **`cognito-identity`**

> Una AWS risorsa con servizi di autenticazione e autorizzazione per applicazioni che utilizzano credenziali temporanee. AWS

Identità non autenticata

> Un utente che non ha effettuato l'accesso con un IdP del pool di identità. Puoi consentire agli utenti di generare credenziali utente limitate per un singolo ruolo IAM prima dell'autenticazione.

Credenziali utente

> Chiavi AWS API temporanee che gli utenti ricevono dopo l'autenticazione del pool di identità.

Utilizzo di questo servizio con un SDK AWS

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

> **Esempio di disponibilità**
>
> Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Iniziare con AWS

Prima di iniziare a lavorare con Amazon Cognito, disponi di alcune risorse necessarie. AWS

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Nozioni di base sui bacini d'utenza

È possibile utilizzare le guide in questa sezione per creare le risorse iniziali del pool di utenti. Per una step-by-step guida dettagliata, inizia con un'applicazione web di base nell'ambiente di JavaScript sviluppo React. Da lì, puoi continuare ad aggiungere funzionalità come l'interfaccia utente ospitata (interfaccia utente ospitata) e l'accesso federato con social network o provider di identità SAML 2.0 esterni (). IdPs

Mentre lavori per espandere il tuo set di funzionalità e incorporare più componenti di Amazon Cognito, leggi il capitolo sui pool di utenti di Amazon Cognito per le descrizioni complete di tutto ciò che puoi fare con i pool di utenti.

L'esempio di pool di utenti e applicazione in questa sezione dimostra un'integrazione di base delle risorse applicative con i pool di utenti di Amazon Cognito. Successivamente, puoi modificare il tuo pool di utenti per utilizzare più opzioni disponibili. Quindi puoi aggiornare l'applicazione per adottare nuove API e interagire con l'interfaccia utente ospitata e IdPs.

Il tutorial in questa sezione crea un'applicazione con un'interfaccia utente personalizzata e un'autenticazione basata su API con un SDK. AWS Le applicazioni create in questo modo sono ideali per autenticare gli utenti locali. Per iniziare con un'applicazione con un'interfaccia utente predefinita, la gestione automatica di alcune funzionalità del pool di utenti e l'autenticazione degli utenti federati, vai avanti a. Aggiungi un client di app con l'interfaccia utente ospitata

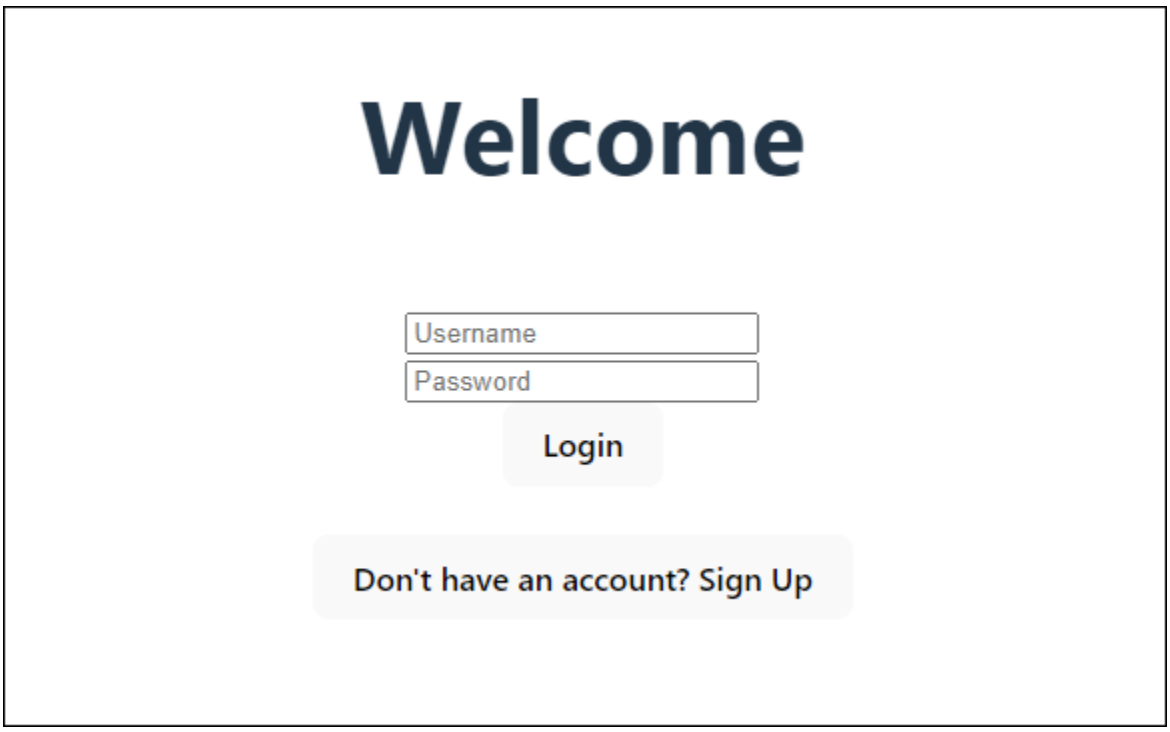
Argomenti

- Configura un esempio di applicazione React a pagina singola
- Configura un'app Android di esempio con Flutter
- Passaggi successivi

Configura un esempio di applicazione React a pagina singola

In questo tutorial, creerai un'applicazione React a pagina singola in cui puoi testare la registrazione, la conferma e l'accesso degli utenti. React è una libreria JavaScript basata su app web e mobili, con particolare attenzione all'interfaccia utente (UI). Questa applicazione di esempio illustra alcune funzioni di base dei pool di utenti di Amazon Cognito. Se hai già esperienza nello sviluppo di app web con React, scarica l'app di esempio da. GitHub

La schermata seguente mostra la pagina di autenticazione iniziale dell'applicazione che creerai.



La procedura Crea un pool di utenti consente di impostare un pool di utenti che funziona con l'applicazione di esempio. È possibile saltare questo passaggio se si dispone di un pool di utenti che soddisfa i seguenti requisiti:

- Gli utenti possono accedere con il proprio indirizzo e-mail. Opzioni di accesso al pool di utenti di Cognito: e-mail.
- I nomi utente non fanno differenza tra maiuscole e minuscole. Requisiti relativi al nome utente: l'opzione Fai distinzione tra maiuscole e minuscole non è selezionata.
- L'autenticazione a più fattori (MFA) non è richiesta. Applicazione della MFA: MFA opzionale.
- Il pool di utenti verifica gli attributi per la conferma del profilo utente con un messaggio di posta elettronica. Attributi da verificare: invio di un messaggio e-mail, verifica dell'indirizzo e-mail.
- L'email è l'unico attributo obbligatorio. Attributi obbligatori: email.
- Gli utenti possono registrarsi nel tuo pool di utenti. Autoregistrazione: è selezionata l'opzione Abilita l'autoregistrazione.
- Il client iniziale dell'app è un client pubblico che consente l'accesso con nome utente e password. Tipo di app: Client pubblico, Flussi di autenticazione:. `ALLOW_USER_PASSWORD_AUTH`

Creazione di un pool di utenti

Crea un nuovo pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli il pulsante Crea pool di utenti. Potrebbe esserti richiesto di selezionare User Pools dal riquadro di navigazione a sinistra per visualizzare questa opzione.
3. Nell'angolo in alto a destra della pagina, scegli Create a User Pool (Crea bacino d'utenza).
4. In Configura l'esperienza di accesso, puoi scegliere i provider di identità (IdPs) che utilizzerai con questo pool di utenti. Per ulteriori informazioni, consulta Aggiunta di un accesso al bacino d'utenza tramite terze parti.
    a. In Provider di autenticazione, per i tipi di provider, assicurati che sia selezionato solo il pool di utenti di Cognito.
    b. Per le opzioni di accesso al pool di utenti di Cognito, scegli Nome utente. Non selezionate alcun requisito aggiuntivo per il nome utente.
    c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
5. In Configura i requisiti di sicurezza, puoi scegliere la politica delle password, i requisiti di autenticazione a più fattori (MFA) e le opzioni di ripristino dell'account utente. Per ulteriori informazioni, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.
    a. Per i criteri relativi alle password, verifica che la modalità politica delle password sia impostata sui valori predefiniti di Cognito.
    b. In Autenticazione a più fattori, per l'applicazione della MFA, scegli MFA opzionale.
    c. Per i metodi MFA, scegli App di autenticazione e messaggi SMS.
    d. Per il ripristino dell'account utente, conferma che sia selezionata l'opzione Abilita il ripristino dell'account in modalità self-service e che il metodo di recapito dei messaggi di ripristino dell'account utente sia impostato su Solo e-mail.
    e. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
6. In Configura l'esperienza di registrazione, puoi determinare in che modo i nuovi utenti verificheranno la propria identità al momento della registrazione come nuovo utente e quali attributi devono essere obbligatori o facoltativi durante il flusso di registrazione degli utenti. Per ulteriori informazioni, consulta Gestione degli utenti nel tuo bacino d'utenza.

a. Conferma che l'opzione Abilita l'autoregistrazione sia selezionata. Questa impostazione apre il tuo pool di utenti alla registrazione di chiunque su Internet. Questa impostazione è destinata agli scopi dell'applicazione di esempio, ma applica questa impostazione con cautela negli ambienti di produzione.

b. In Verifica e conferma assistite da Cognito, verifica che la casella di controllo Consenti a Cognito di inviare automaticamente messaggi per verificare e confermare sia selezionata.

c. Conferma che gli Attributi da verificare siano impostati su Invia messaggio e-mail, verifica indirizzo e-mail.

d. In Verifica delle modifiche agli attributi, conferma che siano selezionate le opzioni predefinite: Mantieni il valore dell'attributo originale quando un aggiornamento è in sospeso è selezionato e Valori degli attributi attivi quando un aggiornamento è in sospeso è impostato su Indirizzo e-mail.

e. In Attributi obbligatori, verifica che gli attributi richiesti in base alle selezioni precedenti visualizzino l'e-mail.

> **Important**
>
> Per questa applicazione di esempio, il tuo pool di utenti non deve impostare phone_number come attributo obbligatorio. Se phone_number viene mostrato come attributo obbligatorio, rivedi e aggiorna le tue scelte precedenti:
>
> - MFA opzionale, solo e-mail per il metodo di consegna per i messaggi di ripristino dell'account utente
> - Invia messaggio e-mail, verifica l'indirizzo e-mail per la verifica degli attributi

f. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

7. In Configura il recapito dei messaggi, puoi configurare l'integrazione con Amazon Simple Email Service e Amazon Simple Notification Service per inviare messaggi e-mail e SMS ai tuoi utenti per la registrazione, la conferma dell'account, l'MFA e il ripristino dell'account. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito e Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

   a. Per Provider di posta elettronica, scegli Invia e-mail con Cognito e utilizza il mittente e-mail predefinito fornito da Amazon Cognito. Questa impostazione per un volume di posta elettronica basso è sufficiente per il test delle applicazioni. Puoi effettuare il reso dopo aver

verificato un indirizzo e-mail con Amazon Simple Email Service (Amazon SES) e aver scelto Invia e-mail con Amazon SES.

b. Per gli SMS, seleziona Crea un nuovo ruolo IAM e inserisci il nome del ruolo IAM. Questo crea un ruolo che concede le autorizzazioni ad Amazon Cognito per inviare messaggi SMS.

c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

8. In Integra la tua app, puoi assegnare un nome al tuo pool di utenti, configurare l'interfaccia utente ospitata e creare un client per l'app. Per ulteriori informazioni, consulta Aggiungi un client di app con l'interfaccia utente ospitata. Le applicazioni di esempio non utilizzano l'interfaccia utente ospitata.

   a. In Nome del pool di utenti, inserisci un nome del pool di utenti.

   b. Non selezionare Usa l'interfaccia utente ospitata da Cognito.

   c. In Client iniziale dell'app, verifica che il tipo di app sia impostato su Client pubblico.

   d. In Client secret, conferma che sia selezionata l'opzione Non generare un segreto client.

   e. Inserisci un nome del client dell'App.

   f. Espandi le impostazioni avanzate del client dell'app. Aggiungi `ALLOW_USER_PASSWORD_AUTH` all'elenco dei flussi di autenticazione.

   g. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

9. Rivedi le tue scelte nella schermata Rivedi e crea e modifica le selezioni in base alle esigenze. Quando sei soddisfatto della configurazione del pool di utenti, scegli Crea pool di utenti per procedere.

10. Dalla pagina Pool di utenti, scegli il tuo nuovo pool di utenti.

11. Nella sezione Panoramica del pool di utenti, annota l'ID del tuo pool di utenti. Fornirai questa stringa quando creerai l'applicazione di esempio.

12. Scegli la scheda Integrazione delle app e individua la sezione Client e analisi delle app. Seleziona il tuo nuovo client per l'app. Annota il tuo ID cliente.

Risorse correlate

- Bacini d'utenza di Amazon Cognito
- Flusso di autenticazione del bacino d'utenza
- Utilizzo di token con bacini d'utenza

Creazione di un'applicazione

Per creare questa applicazione, è necessario configurare un ambiente di sviluppo. I requisiti dell'ambiente di sviluppo sono:

1. Node.js è installato e aggiornato.
2. Node package manager (npm) è installato e aggiornato almeno alla versione 10.2.3.
3. L'ambiente è accessibile sulla porta TCP 5173 in un browser web.

Per creare un esempio di applicazione web React

1. Accedi al tuo ambiente di sviluppo e vai alla directory principale dell'applicazione.

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

2. Crea un nuovo servizio React.

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

3. Clona la cartella del `cognito-developer-guide-react-example` progetto dal repository degli esempi di AWS codice in poi. 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. Vai alla `src` directory del tuo progetto.

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

5. Modifica `config.ts` e sostituisci i seguenti valori:

   a. Sostituisci `YOUR_AWS_REGION` con un Regione AWS codice. Ad esempio: `us-east-1`.

b. Sostituiscilo `YOUR_COGNITO_USER_POOL_ID` con l'ID del pool di utenti che hai designato per il test. Ad esempio: `us-east-1_EXAMPLE`. Il pool di utenti deve appartenere a Regione AWS quello inserito nel passaggio precedente.

c. Sostituiscilo `YOUR_COGNITO_APP_CLIENT_ID` con l'ID del client dell'app che hai designato per il test. Ad esempio: `1example23456789`. Il client dell'app deve essere incluso nel pool di utenti del passaggio precedente.

6. Se desideri accedere all'applicazione di esempio da un IP diverso da`localhost`, modifica `package.json` e modifica la riga `"dev": "vite",` in`"dev": "vite --host 0.0.0.0",`.
7. Installa la tua applicazione.

```
npm install
```

8. Avvia l'applicazione.

```
npm run dev
```

9. Accedere all'applicazione in un browser Web all'indirizzo `http://localhost:5173` o`http://[IP address]:5173`.
10. Registra un nuovo utente con un indirizzo email valido.
11. Recupera il codice di conferma dal tuo messaggio e-mail. Inserisci il codice di conferma nell'applicazione.
12. Accedi con il tuo nome utente e la tua password.

Creazione di un ambiente di sviluppo React con Amazon Lightsail

Un modo rapido per iniziare a usare questa applicazione è creare un server cloud virtuale con Amazon Lightsail.

Con Lightsail, puoi creare rapidamente una piccola istanza di server preconfigurata con i prerequisiti per questa applicazione di esempio. Puoi accedere tramite SSH alla tua istanza con un client basato su browser e connetterti al server web con un indirizzo IP pubblico o privato.

Per creare un'istanza Lightsail per questa applicazione di esempio

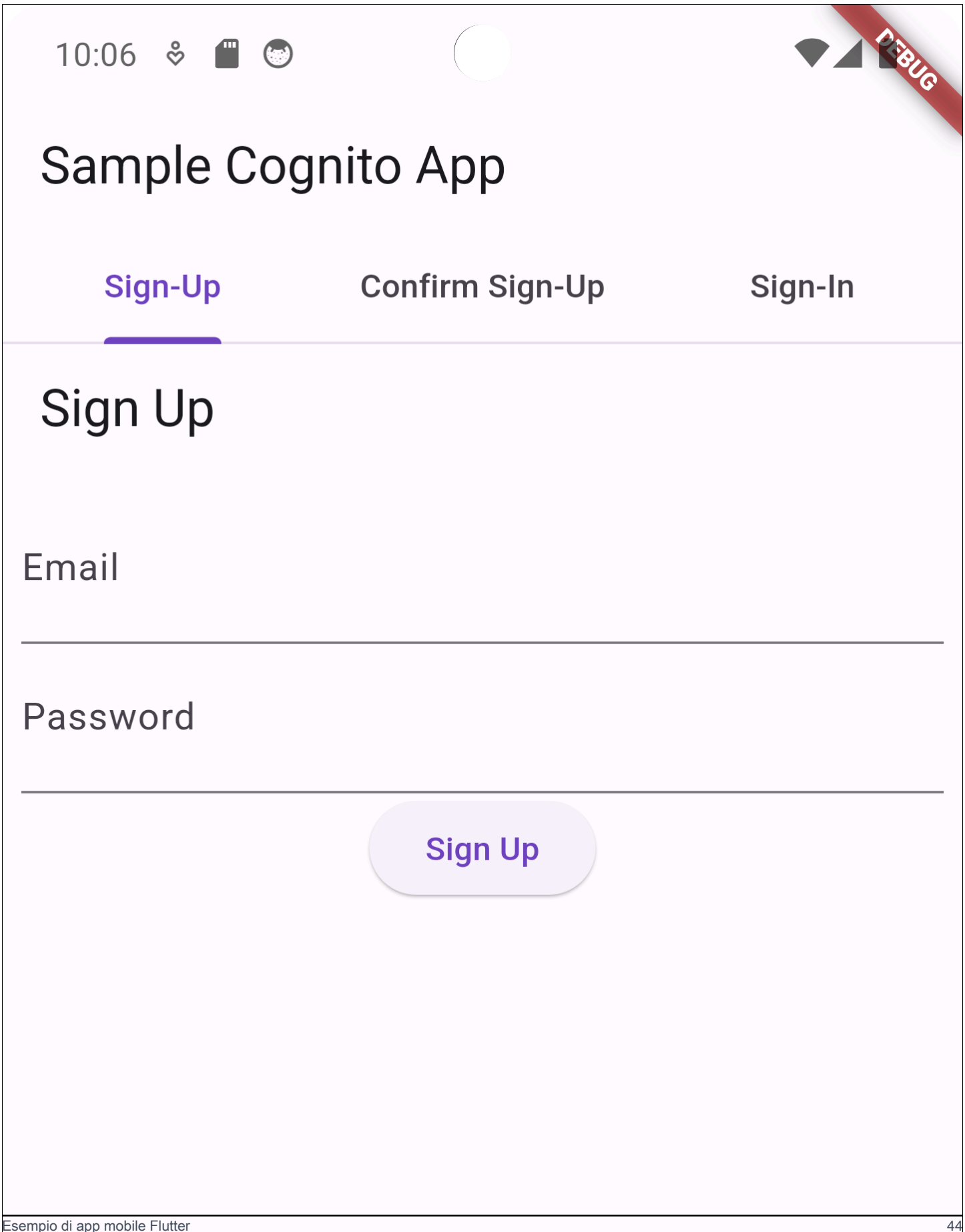
1. Vai alla console Lightsail. Se richiesto, inserisci le tue credenziali. AWS
2. Seleziona Crea istanza.

3. Per Seleziona una piattaforma, scegli Linux/Unix.
4. Per Seleziona un progetto, scegli Node.js.
5. In Identifica la tua istanza, assegna un nome descrittivo al tuo ambiente di sviluppo.
6. Seleziona Crea istanza.
7. Dopo che Lightsail ha creato l'istanza, selezionala e nella scheda Connetti scegli Connetti tramite SSH.
8. Una sessione SSH si apre in una finestra del browser. Esegui `node -v` e `npm -v` conferma che la tua istanza è stata fornita con Node.js e la versione minima di npm 10.2.3.
9. Procedi con la configurazione dell'applicazione React.

Configura un'app Android di esempio con Flutter

In questo tutorial, creerai un'applicazione mobile in Android Studio in cui potrai emulare un dispositivo e testare la registrazione, la conferma e l'accesso degli utenti. Questa applicazione di esempio crea un client mobile di base per pool di utenti Amazon Cognito per Android in Flutter. Se hai già esperienza nello sviluppo di app per dispositivi mobili con Flutter, scarica l'app di esempio da. GitHub

La schermata seguente mostra l'app in esecuzione su un dispositivo Android virtuale.


10:06
DEBUG
Sample Cognito App
Sign-Up
Confirm Sign-Up
Sign-In
Sign Up
Email
Password
Sign Up

La procedura Crea un pool di utenti consente di impostare un pool di utenti che funziona con l'applicazione di esempio. È possibile saltare questo passaggio se si dispone di un pool di utenti che soddisfa i seguenti requisiti:

- Gli utenti possono accedere con il proprio indirizzo e-mail. Opzioni di accesso al pool di utenti di Cognito: e-mail.
- I nomi utente non fanno differenza tra maiuscole e minuscole. Requisiti relativi al nome utente: l'opzione Fai distinzione tra maiuscole e minuscole non è selezionata.
- L'autenticazione a più fattori (MFA) non è richiesta. Applicazione della MFA: MFA opzionale.
- Il pool di utenti verifica gli attributi per la conferma del profilo utente con un messaggio di posta elettronica. Attributi da verificare: invio di un messaggio e-mail, verifica dell'indirizzo e-mail.
- L'email è l'unico attributo obbligatorio. Attributi obbligatori: email.
- Gli utenti possono registrarsi nel tuo pool di utenti. Autoregistrazione: è selezionata l'opzione Abilita l'autoregistrazione.
- Il client iniziale dell'app è un client pubblico che consente l'accesso con nome utente e password. Tipo di app: Client pubblico, Flussi di autenticazione:. `ALLOW_USER_PASSWORD_AUTH`

Creazione di un pool di utenti

Crea un nuovo pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli il pulsante Crea pool di utenti. Potrebbe esserti richiesto di selezionare User Pools dal riquadro di navigazione a sinistra per visualizzare questa opzione.
3. Nell'angolo in alto a destra della pagina, scegli Create a User Pool (Crea bacino d'utenza).
4. In Configura l'esperienza di accesso, puoi scegliere i provider di identità (IdPs) che utilizzerai con questo pool di utenti. Per ulteriori informazioni, consulta Aggiunta di un accesso al bacino d'utenza tramite terze parti.
    a. In Provider di autenticazione, per i tipi di provider, assicurati che sia selezionato solo il pool di utenti di Cognito.
    b. Per le opzioni di accesso al pool di utenti di Cognito, scegli Nome utente. Non selezionate alcun requisito aggiuntivo per il nome utente.
    c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

5. In Configura i requisiti di sicurezza, puoi scegliere la politica delle password, i requisiti di autenticazione a più fattori (MFA) e le opzioni di ripristino dell'account utente. Per ulteriori informazioni, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.

   a. Per i criteri relativi alle password, verifica che la modalità politica delle password sia impostata sui valori predefiniti di Cognito.

   b. In Autenticazione a più fattori, per l'applicazione della MFA, scegli MFA opzionale.

   c. Per i metodi MFA, scegli App di autenticazione e messaggi SMS.

   d. Per il ripristino dell'account utente, conferma che sia selezionata l'opzione Abilita il ripristino dell'account in modalità self-service e che il metodo di recapito dei messaggi di ripristino dell'account utente sia impostato su Solo e-mail.

   e. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

6. In Configura l'esperienza di registrazione, puoi determinare in che modo i nuovi utenti verificheranno la propria identità al momento della registrazione come nuovo utente e quali attributi devono essere obbligatori o facoltativi durante il flusso di registrazione degli utenti. Per ulteriori informazioni, consulta Gestione degli utenti nel tuo bacino d'utenza.

   a. Conferma che l'opzione Abilita l'autoregistrazione sia selezionata. Questa impostazione apre il tuo pool di utenti alla registrazione di chiunque su Internet. Questa impostazione è destinata agli scopi dell'applicazione di esempio, ma applica questa impostazione con cautela negli ambienti di produzione.

   b. In Verifica e conferma assistite da Cognito, verifica che la casella di controllo Consenti a Cognito di inviare automaticamente messaggi per verificare e confermare sia selezionata.

   c. Conferma che gli Attributi da verificare siano impostati su Invia messaggio e-mail, verifica indirizzo e-mail.

   d. In Verifica delle modifiche agli attributi, conferma che siano selezionate le opzioni predefinite: Mantieni il valore dell'attributo originale quando un aggiornamento è in sospeso è selezionato e Valori degli attributi attivi quando un aggiornamento è in sospeso è impostato su Indirizzo e-mail.

   e. In Attributi obbligatori, verifica che gli attributi richiesti in base alle selezioni precedenti visualizzino l'e-mail.

> **Important**
>
> Per questa applicazione di esempio, il tuo pool di utenti non deve impostare phone_number come attributo obbligatorio. Se phone_number viene mostrato come attributo obbligatorio, rivedi e aggiorna le tue scelte precedenti:
>
> - MFA opzionale, solo e-mail per il metodo di consegna per i messaggi di ripristino dell'account utente
> - Invia messaggio e-mail, verifica l'indirizzo e-mail per la verifica degli attributi

f. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

7. In Configura il recapito dei messaggi, puoi configurare l'integrazione con Amazon Simple Email Service e Amazon Simple Notification Service per inviare messaggi e-mail e SMS ai tuoi utenti per la registrazione, la conferma dell'account, l'MFA e il ripristino dell'account. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito e Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

   a. Per Provider di posta elettronica, scegli Invia e-mail con Cognito e utilizza il mittente e-mail predefinito fornito da Amazon Cognito. Questa impostazione per un volume di posta elettronica basso è sufficiente per il test delle applicazioni. Puoi effettuare il reso dopo aver verificato un indirizzo e-mail con Amazon Simple Email Service (Amazon SES) e aver scelto Invia e-mail con Amazon SES.

   b. Per gli SMS, seleziona Crea un nuovo ruolo IAM e inserisci il nome del ruolo IAM. Questo crea un ruolo che concede le autorizzazioni ad Amazon Cognito per inviare messaggi SMS.

   c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

8. In Integra la tua app, puoi assegnare un nome al tuo pool di utenti, configurare l'interfaccia utente ospitata e creare un client per l'app. Per ulteriori informazioni, consulta Aggiungi un client di app con l'interfaccia utente ospitata. Le applicazioni di esempio non utilizzano l'interfaccia utente ospitata.

   a. In Nome del pool di utenti, inserisci un nome del pool di utenti.

   b. Non selezionare Usa l'interfaccia utente ospitata da Cognito.

   c. In Client iniziale dell'app, verifica che il tipo di app sia impostato su Client pubblico.

   d. In Client secret, conferma che sia selezionata l'opzione Non generare un segreto client.

   e. Inserisci un nome del client dell'App.

f. Espandi le impostazioni avanzate del client dell'app. Aggiungi `ALLOW_USER_PASSWORD_AUTH` all'elenco dei flussi di autenticazione.

g. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

9. Rivedi le tue scelte nella schermata Rivedi e crea e modifica le selezioni in base alle esigenze. Quando sei soddisfatto della configurazione del pool di utenti, scegli Crea pool di utenti per procedere.
10. Dalla pagina Pool di utenti, scegli il tuo nuovo pool di utenti.
11. Nella sezione Panoramica del pool di utenti, annota l'ID del tuo pool di utenti. Fornirai questa stringa quando creerai l'applicazione di esempio.
12. Scegli la scheda Integrazione delle app e individua la sezione Client e analisi delle app. Seleziona il tuo nuovo client per l'app. Annota il tuo ID cliente.

Risorse correlate

- Bacini d'utenza di Amazon Cognito
- Flusso di autenticazione del bacino d'utenza
- Utilizzo di token con bacini d'utenza

Creazione di un'applicazione

Per creare un'app Android di esempio

1. Installa Android Studio e gli strumenti da riga di comando.
2. In Android Studio, installa il plug-in Flutter.
3. Crea un nuovo progetto Android Studio dal contenuto della `cognito_flutter_mobile_app` directory in questa app di esempio.

   - Modifica `assets/config.json` e sostituisci `<<YOUR USER POOL ID>>` e `<< YOUR CLIENT ID>>` con gli ID del pool di utenti e del client dell'app che hai creato in precedenza.

4. Installa Flutter.

   a. Aggiungi Flutter alla tua variabile PATH.

   b. Accetta le licenze con il seguente comando.

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

c. Verifica il tuo ambiente Flutter e installa tutti i componenti mancanti.

```
flutter doctor
```

- Se mancano dei componenti, esegui `flutter doctor -v` per scoprire come risolvere il problema.

d. Passa alla directory del tuo nuovo progetto Flutter e installa le dipendenze.

- Esegui `flutter pub add amazon_cognito_identity_dart_2`.

e. Esegui `flutter pub add flutter_secure_storage`.

5. Crea un dispositivo Android virtuale.

1. Nella GUI di Android Studio, crea un nuovo dispositivo con il gestore dispositivi.
2. Nella CLI, esegui. `flutter emulators --create --name android-device`

6. Avvia il tuo dispositivo Android virtuale.

1. Nella GUI di Android Studio, seleziona l'  icona di avvio accanto al tuo dispositivo virtuale.
2. Nella CLI, esegui. `flutter emulators --launch android-device`

7. Avvia l'app sul tuo dispositivo virtuale.

1. Nella GUI di Android Studio, seleziona l'icona di distribuzione.

2. Nella CLI, esegui. `flutter run`

8. Accedi al tuo dispositivo virtuale in esecuzione in Android Studio.
9. Registra un nuovo utente con un indirizzo email valido.
10. Recupera il codice di conferma dal tuo messaggio e-mail. Inserisci il codice di conferma nell'applicazione.
11. Accedi con il tuo nome utente e la tua password.

Passaggi successivi

Dopo aver seguito i tutorial per completare applicazioni di esempio, puoi ampliare l'ambito dell'implementazione del tuo pool di utenti. È possibile creare pool di utenti aggiuntivi, personalizzare le funzionalità del pool di utenti per altre applicazioni o aggiungere provider di identità esterni. Mentre pianifichi il passaggio ai pool di utenti di Amazon Cognito nelle applicazioni di produzione, puoi valutare esempi e tutorial aggiuntivi.

Di seguito sono riportate alcune funzionalità aggiuntive dei pool di utenti di Amazon Cognito:

- Personalizzazione delle pagine Web di registrazione e accesso integrate
- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Aggiunta di sicurezza avanzata a un bacino d'utenza
- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Per una panoramica dei modelli di autenticazione e autorizzazione di Amazon Cognito, consulta. Come funziona l'autenticazione con i pool di utenti e i pool di identità di Amazon Cognito

Per accedere ad altri utenti Servizi AWS dopo una corretta autenticazione del pool di utenti, consultaAccesso Servizi AWS tramite un pool di identità dopo l'accesso.

Oltre a utilizzare gli SDK AWS Management Console e il pool di utenti, puoi anche gestire i tuoi pool di utenti utilizzando il AWS Command Line Interface.

Argomenti

- Crea un nuovo pool di utenti
- Aggiungi un client di app con l'interfaccia utente ospitata
- Aggiunta di un accesso social a un pool di utenti (facoltativo)
- Aggiunta di un accesso con un provider di identità SAML a un bacino d'utenza (facoltativo)

Crea un nuovo pool di utenti

Con un bacino d'utenza, gli utenti possono accedere all'app Web o mobile tramite Amazon Cognito.

Crea un nuovo pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli il pulsante Crea pool di utenti. Potrebbe esserti richiesto di selezionare User Pools dal riquadro di navigazione a sinistra per visualizzare questa opzione.
3. Nell'angolo in alto a destra della pagina, scegli Create a User Pool (Crea bacino d'utenza).
4. In Configura l'esperienza di accesso, puoi scegliere i provider di identità (IdPs) che utilizzerai con questo pool di utenti. Per ulteriori informazioni, consulta Aggiunta di un accesso al bacino d'utenza tramite terze parti.
    a. In Provider di autenticazione, per i tipi di provider, assicurati che sia selezionato solo il pool di utenti di Cognito.
    b. Per le opzioni di accesso al pool di utenti di Cognito, scegli Nome utente. Non selezionate alcun requisito aggiuntivo per il nome utente.
    c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
5. In Configura i requisiti di sicurezza, puoi scegliere la politica delle password, i requisiti di autenticazione a più fattori (MFA) e le opzioni di ripristino dell'account utente. Per ulteriori informazioni, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.
    a. Per i criteri relativi alle password, verifica che la modalità politica delle password sia impostata sui valori predefiniti di Cognito.
    b. In Autenticazione a più fattori, per l'applicazione della MFA, scegli MFA opzionale.
    c. Per i metodi MFA, scegli App di autenticazione e messaggi SMS.
    d. Per il ripristino dell'account utente, conferma che sia selezionata l'opzione Abilita il ripristino dell'account in modalità self-service e che il metodo di recapito dei messaggi di ripristino dell'account utente sia impostato su Solo e-mail.
    e. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
6. In Configura l'esperienza di registrazione, puoi determinare in che modo i nuovi utenti verificheranno la propria identità al momento della registrazione come nuovo utente e quali attributi devono essere obbligatori o facoltativi durante il flusso di registrazione degli utenti. Per ulteriori informazioni, consulta Gestione degli utenti nel tuo bacino d'utenza.
    a. Conferma che l'opzione Abilita l'autoregistrazione sia selezionata. Questa impostazione apre il tuo pool di utenti alla registrazione di chiunque su Internet. Questa impostazione

è destinata agli scopi dell'applicazione di esempio, ma applica questa impostazione con cautela negli ambienti di produzione.

b. In Verifica e conferma assistite da Cognito, verifica che la casella di controllo Consenti a Cognito di inviare automaticamente messaggi per verificare e confermare sia selezionata.

c. Conferma che gli Attributi da verificare siano impostati su Invia messaggio e-mail, verifica indirizzo e-mail.

d. In Verifica delle modifiche agli attributi, conferma che siano selezionate le opzioni predefinite: Mantieni il valore dell'attributo originale quando un aggiornamento è in sospeso è selezionato e Valori degli attributi attivi quando un aggiornamento è in sospeso è impostato su Indirizzo e-mail.

e. In Attributi obbligatori, verifica che gli attributi richiesti in base alle selezioni precedenti visualizzino l'e-mail.

> **Important**
>
> Per questa applicazione di esempio, il tuo pool di utenti non deve impostare phone_number come attributo obbligatorio. Se phone_number viene mostrato come attributo obbligatorio, rivedi e aggiorna le tue scelte precedenti:
>
> - MFA opzionale, solo e-mail per il metodo di consegna per i messaggi di ripristino dell'account utente
> - Invia messaggio e-mail, verifica l'indirizzo e-mail per la verifica degli attributi

f. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.

7. In Configura il recapito dei messaggi, puoi configurare l'integrazione con Amazon Simple Email Service e Amazon Simple Notification Service per inviare messaggi e-mail e SMS ai tuoi utenti per la registrazione, la conferma dell'account, l'MFA e il ripristino dell'account. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito e Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

   a. Per Provider di posta elettronica, scegli Invia e-mail con Cognito e utilizza il mittente e-mail predefinito fornito da Amazon Cognito. Questa impostazione per un volume di posta elettronica basso è sufficiente per il test delle applicazioni. Puoi effettuare il reso dopo aver verificato un indirizzo e-mail con Amazon Simple Email Service (Amazon SES) e aver scelto Invia e-mail con Amazon SES.

   b. Per gli SMS, seleziona Crea un nuovo ruolo IAM e inserisci il nome del ruolo IAM. Questo crea un ruolo che concede le autorizzazioni ad Amazon Cognito per inviare messaggi SMS.
   c. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
8. In Integra la tua app, puoi assegnare un nome al tuo pool di utenti, configurare l'interfaccia utente ospitata e creare un client per l'app. Per ulteriori informazioni, consulta Aggiungi un client di app con l'interfaccia utente ospitata. Le applicazioni di esempio non utilizzano l'interfaccia utente ospitata.

   a. In Nome del pool di utenti, inserisci un nome del pool di utenti.
   b. Non selezionare Usa l'interfaccia utente ospitata da Cognito.
   c. In Client iniziale dell'app, verifica che il tipo di app sia impostato su Client pubblico.
   d. In Client secret, conferma che sia selezionata l'opzione Non generare un segreto client.
   e. Inserisci un nome del client dell'App.
   f. Espandi le impostazioni avanzate del client dell'app. Aggiungi `ALLOW_USER_PASSWORD_AUTH` all'elenco dei flussi di autenticazione.
   g. Mantieni tutte le altre opzioni come predefinite e scegli Avanti.
9. Rivedi le tue scelte nella schermata Rivedi e crea e modifica le selezioni in base alle esigenze. Quando sei soddisfatto della configurazione del pool di utenti, scegli Crea pool di utenti per procedere.
10. Dalla pagina Pool di utenti, scegli il tuo nuovo pool di utenti.
11. Nella sezione Panoramica del pool di utenti, annota l'ID del tuo pool di utenti. Fornirai questa stringa quando creerai l'applicazione di esempio.
12. Scegli la scheda Integrazione delle app e individua la sezione Client e analisi delle app. Seleziona il tuo nuovo client per l'app. Annota il tuo ID cliente.

Per creare un bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (bacini d'utenza).
3. Nell'angolo in alto a destra della pagina, scegli Create a User Pool (Crea bacino d'utenza) per avviare la procedura guidata di creazione del bacino d'utenza.
4. Alla voce Configure sign-in experience (configura l'esperienza di accesso), scegli i provider federati che desideri utilizzare con questo bacino d'utenza. Per ulteriori informazioni, consulta Aggiunta di un accesso al bacino d'utenza tramite terze parti.

5. Alla voce Configure security requirements (configurazione dei requisiti di sicurezza), selezionaere la policy della password, i requisiti di autenticazione a più fattori (MFA) e le opzioni di ripristino dell'account utente. Per ulteriori informazioni, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.
6. Alla voce Configure sign-up experience (configurazione dell'esperienza di registrazione), determinare in che modo i nuovi utenti verificheranno le loro identità al momento della registrazione e quali attributi devono essere necessari o facoltativi durante il flusso di registrazione dell'utente. Per ulteriori informazioni, consulta Gestione degli utenti nel tuo bacino d'utenza.

   > **Important**
   >
   > Se attivi la registrazione dell'utente nel pool di utenti, chiunque su Internet può effettuare la registrazione a un account e accedere alle tue app. Non abilitare la registrazione self-service nel pool di utenti a meno che non desideri aprire l'app alla registrazione pubblica. Per modificare questa impostazione, aggiorna l'iscrizione in modalità self-service nella scheda Esperienza di registrazione della console del pool di utenti o aggiorna il valore di AllowAdminCreateUserOnlyin una richiesta o API. CreateUserPool UpdateUserPool Per informazioni sulle funzionalità di sicurezza che puoi configurare nei pool di utenti, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.

7. Alla voce Configure message delivery (configurazione della consegna di messaggi), configurare l'integrazione con Amazon Simple Email Service e Amazon Simple Notification Service per inviare e-mail e messaggi SMS agli utenti per la registrazione, la conferma dell'account, l'MFA e il recupero dell'account. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito e Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.
8. Alla voce Integrate your app (Integrazione dell'app), assegnare un nome al bacino d'utenza, configurare l'interfaccia utente ospitata e creare un client app. Per ulteriori informazioni, consulta Aggiungi un client di app con l'interfaccia utente ospitata.
9. Controlla le tue scelte nella schermata Rivedi e crea e modifica le selezioni in base alle tue esigenze. Quando sei soddisfatto della configurazione del pool di utenti, seleziona Crea pool di utenti per procedere.

Risorse correlate

Per ulteriori informazioni sui bacini d'utenza, consultare Bacini d'utenza di Amazon Cognito.

Vedi anche: Flusso di autenticazione del bacino d'utenza eUtilizzo di token con bacini d'utenza.

Aggiungi un client di app con l'interfaccia utente ospitata

Dopo aver creato un pool di utenti, puoi creare un client di app per un'applicazione che visualizza le pagine Web integrate dell'interfaccia utente ospitata. Nell'interfaccia utente ospitata, gli utenti possono:

- Registrarsi per creare un profilo utente.
- Accedi con un provider di identità di terze parti.
- Accedi con o senza autenticazione a più fattori.
- Reimposta la loro password.

Per creare un client di app per l'accesso all'interfaccia utente ospitata

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo. Se crei un nuovo bacino d'utenza, ti verrà richiesto di configurare un client app e configurare l'interfaccia utente ospitata durante la procedura guidata.
4. Passare alla scheda App Integration (Integrazione App) del tuo bacino d'utenza.
5. Accanto a Dominio, scegli Operazioni e quindi Crea dominio personalizzato o Crea dominio Cognito. Se hai già configurato un dominio del pool di utenti, scegli Elimina dominio Amazon Cognito o Elimina dominio personalizzato prima di creare il nuovo dominio personalizzato.
6. Inserisci un prefisso dominio disponibile da utilizzare con un Dominio Amazon Cognito. Per informazioni sulla configurazione di un dominio personalizzato, consulta la sezione Utilizzo del proprio dominio per l'interfaccia utente ospitata.
7. Scegli Create (Crea).
8. Torna alla scheda integrazione app per lo stesso bacino d'utenza e individua l'opzione client dell'app. Scegli Create app client (crea client dell'app).
9. Scegli un tipo di applicazione. Verranno fornite alcune impostazioni consigliate in base alla tua selezione. Un'app che utilizza l'interfaccia utente ospitata è un client pubblico.

10. Insrerisci un nome del client dell'App.
11. Per questo esercizio, scegli l'opzione non generare il segreto client. Il segreto client viene utilizzato dalle app riservate che autenticano gli utenti da un'applicazione centralizzata. In questo esercizio, presenterai una pagina di accesso all'interfaccia utente ospitata agli utenti e non richiederà un segreto client.
12. Scegli i flussi di autenticazione che consentirai con la tua app. Assicurati che `USER_SRP_AUTH` sia stato selezionato.
13. Personalizza la scadenza token, la configurazione avanzata della sicurezza e le autorizzazioni di lettura e scrittura degli attributi a seconda delle esigenze. Per ulteriori informazioni, consulta la sezione configurazione delle impostazioni del client dell'app.
14. Aggiungi un URL di callback per il client dell'app. Qui verrai indirizzato dopo l'autenticazione dell'interfaccia utente ospitata. Non è necessario aggiungere un URL di disconnessione consentito finché non sarai in grado di implementare la disconnessione nella tua app.

    Per un'app iOS o Android, puoi utilizzare un URL di callback del tipo `myapp://`.
15. Seleziona provider di identità per il client dell'app. Come minimo, abilita il pool di utenti di Amazon Cognito come un provider.

    > **Note**
    >
    > Per accedere con provider di identità esterni (IdPs) come Facebook, Amazon, Google e Apple, nonché tramite OpenID Connect (OIDC) o SAML IdPs, configurali innanzitutto come mostrato in Aggiungere l'accesso al pool di utenti tramite una terza parte. Quindi torna alla pagina delle impostazioni del client dell'app per abilitarle.
16. Scegli i Tipi di concessione OAuth 2.0. Seleziona Authorization code grant (Concessione del codice di autorizzazione) per restituire un codice di autorizzazione che viene quindi scambiato per i token dei bacini d'utenza. Poiché i token non vengono mai esposti direttamente a un utente finale, vi sono meno probabilità che vengano compromessi. Tuttavia, per un'applicazione personalizzata è obbligatorio nel back-end scambiare il codice di autorizzazione per i token dei bacini d'utenza. Per motivi di sicurezza, ti consigliamo di utilizzare il flusso di concessione del codice di autorizzazione insieme al Proof Key for Code Exchange (PKCE) per le app mobili.

    Seleziona Concessione implicita per fare in modo che Amazon Cognito restituisca i token Web JSON (JWT) del bacino d'utenza. È possibile usare questo flusso quando non è disponibile un back-end per scambiare un codice di autorizzazione per i token. È inoltre utile per il debug dei token.

> **Note**
>
> È possibile abilitare sia Authorization code grant (Concessione del codice di autorizzazione) sia Implicit code grant (Concessione implicita del codice) e usare entrambe in base alle esigenze.
> Seleziona Client credentials (Credenziali del client) solo se è necessario che l'app richieda i token di accesso a proprio nome e non a nome di un utente.

17. A meno che non intendi escluderne uno in particolare, seleziona tutti gli ambiti OpenID Connect.
18. Seleziona gli ambiti personalizzati che hai configurato. Gli ambiti personalizzati sono in genere utilizzati con client riservati.
19. Scegli Create (Crea).

Per visualizzare la pagina di accesso

Dalla pagina client dell'app, seleziona Visualizza interfaccia utente ospitata per aprire una nuova scheda del browser in una pagina di accesso precompilata con l'ID client dell'app, l'ambito, la concessione e i parametri URL di callback.

È possibile visualizzare la pagina Web di accesso con l'interfaccia utente ospitata tramite il seguente URL. Prendi nota di `response_type`. In questo caso, response_type=code per la concessione del codice di autenticazione.

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

È possibile visualizzare la pagina Web di accesso con l'interfaccia utente ospitata tramite il seguente URL per la concessione implicita del codice dove response_type=token. Dopo aver eseguito correttamente l'accesso, Amazon Cognito restituisce i token del bacino d'utenza alla barra degli indirizzi del browser Web.

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

I token di identità Web JSON sono disponibili subito dopo il parametro `#idtoken=` nella risposta.

L'URL seguente è un esempio di risposta da una richiesta di concessione implicita. La stringa deli token di identità sarà molto più lunga.

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

I token dei bacini d'utenza di Amazon Cognito vengono firmati utilizzando un algoritmo RS256. Puoi decodificare e verificare i token del pool di utenti utilizzando. AWS Lambda Per ulteriori informazioni, consulta Decodificare e verificare i token Amazon Cognito JWT sul sito Web. AWS GitHub

Il dominio è visualizzato nella pagina Domain name (Nome di dominio). L'ID del client app e l'URL di callback sono visualizzati nella pagina General settings (Impostazioni generali). Se le modifiche apportate nella console non vengono visualizzate immediatamente, attendi qualche minuto, quindi aggiorna il browser.

Aggiunta di un accesso social a un pool di utenti (facoltativo)

È possibile abilitare gli utenti della tua app ad accedere tramite un provider di identità (IdP) social come Facebook, Google, Amazon e Apple. Sia che effettuino l'accesso direttamente o attraverso terze parti, tutti gli utenti hanno un profilo nel bacino d'utenza. Salta questa fase se non desideri aggiungere l'accesso attraverso un provider di identità di accesso social.

Registrazione con un IdP social

Prima di creare un IdP social con Amazon Cognito, è necessario registrare l'applicazione con l'IdP social in modo da ricevere un ID client e un segreto client.

Registrazione di un'app con Facebook

1. Creazione di un account sviluppatore con Facebook.
2. Accedi con le tue credenziali di Facebook.
3. Nel menu My Apps (Le mie app), scegli Create New App (Crea nuova app).

   Se non disponi di un'app Facebook esistente, vedrai un'opzione diversa. Scegli Create App (Crea app).
4. Nella pagina Crea un'app, scegli un caso d'uso per l'app, quindi scegli Avanti.
5. Inserisci un nome per la app Facebook, quindi scegli Crea l'app.

6. Nella barra di navigazione a sinistra, scegli Impostazioni app e quindi Base.
7. Prendi nota di ID app e Segreto app. Li utilizzerai nella sezione successiva.
8. Nella parte inferiore della pagina, scegli + Aggiungi piattaforma.
9. Nella schermata Seleziona piattaforma, seleziona le tue piattaforme, quindi scegli Avanti.
10. Scegli Save changes (Salva modifiche).
11. Per i domini dell'app, inserisci il dominio del bacino d'utenza.

    ```
    https://your_user_pool_domain
    ```

12. Scegli Save changes (Salva modifiche).
13. Dalla barra di navigazione, scegli Prodotti, quindi scegli Configura da Facebook Login.
14. Nel menu Facebook Login Configura, scegli Impostazioni.

    Inserisci il tuo URL di reindirizzamento nel campo Valid OAuth Redirect URIs (URI di reindirizzamento OAuth validi). L'URL di reindirizzamento è costituito dal dominio del pool di utenti con l'`/oauth2/idpresponse`endpoint.

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

15. Scegli Save changes (Salva modifiche).

Registrazione di un'app con Amazon

1. Creazione di un account sviluppatore con Amazon.
2. Accedi con le tue credenziali di Amazon.
3. È necessario creare un profilo di sicurezza Amazon per ricevere l'ID client di Amazon e il segreto client.

   Scegli App e servizi dalla barra di navigazione nella parte superiore della pagina, quindi scegli Login with Amazon.
4. Scegli Create a Security Profile (Crea un profilo di sicurezza).
5. Digita un Security Profile Name (nome profilo sicurezza), una Security Profile Description (descrizione profilo sicurezza) e un Consent Privacy Notice URL (URL consenso informativa privacy).
6. Scegli Save (Salva).

7. Scegli Client ID (ID client) e Client Secret (Segreto client) per mostrare i dati relativi. Li utilizzerai nella sezione successiva.
8. Passa il mouse sull'icona a forma di ingranaggio e scegli Web Settings (Impostazioni Web), quindi Scegli Edit (Modifica).
9. Inserisci il dominio del bacino d'utenza nel campo Allowed Origins (origini consentite).

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

10. Digita il dominio del bacino d'utenza con l'endpoint `/oauth2/idpresponse` in Allowed Return URLs (URL restituiti consentiti).

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

11. Scegli Save Salva.

Registrazione di un'app con Google

Per ulteriori informazioni su OAuth 2.0 nella piattaforma Google Cloud, consulta la sezione relativa all'autenticazione e autorizzazione nella documentazione disponibile nella pagina Google Workspace for Developers.

1. Creazione di un account sviluppatore con Google.
2. Accedi alla console Piattaforma Google Cloud.
3. Nella barra di navigazione in alto, scegli Select a project (Seleziona un progetto). Se hai già un progetto nella piattaforma Google, nel menu viene visualizzato il tuo progetto predefinito.
4. Scegli Nuovo progetto.
5. Immetti un nome per il progetto e scegli Crea.
6. Nella barra di navigazione a sinistra, scegli API e servizi, quindi scegli la schermata di consenso Oauth.
7. Inserisci le informazioni sull'app, un dominio dell'app, i domini autorizzati e le informazioni di contatto dello sviluppatore. I domini autorizzati devono includere `amazoncognito.com` e la radice del dominio personalizzato. Ad esempio: `example.com`. Seleziona Salva e continua.
8. 1. In Ambiti, scegli Aggiungi o rimuovi ambiti, quindi scegli almeno i seguenti ambiti OAuth.

   1. `.../auth/userinfo.email`
   2. `.../auth/userinfo.profile`

3. openid

9. In Test users (Utenti di test), scegli Add users (Aggiungi utenti). Inserisci il tuo indirizzo e-mail e tutti gli altri utenti autorizzati al test, quindi scegli SALVA E CONTINUA.
10. Espandi nuovamente la barra di navigazione a sinistra, scegli API e servizi, quindi scegli Credenziali.
11. Scegli CREATE CREDENTIALS, quindi scegli ID client OAuth.
12. Scegli un Application type (Tipo di applicazione) e assegna un Name (Nome) al client.
13. In JavaScript Origini autorizzate, scegli AGGIUNGI URI. Immetti il dominio del pool di utenti.

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

14. In Authorized redirect URIs (URI di reindirizzamento autorizzati), scegli ADD URI (Aggiungi URI). Inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

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

15. Seleziona CREATE.
16. Memorizza in modo sicuro i valori visualizzati da Google in Il tuo ID client e Il tuo segreto del client. Fornisci questi valori ad Amazon Cognito quando aggiungi un gestore dell'identità digitale (IdP) Google.

Come registrare un'app con Apple

Per ulteriori informazioni sulla configurazione della funzionalità di accesso con Apple, consulta la sezione relativa alla configurazione dell'ambiente per la funzionalità di accesso con Apple nella documentazione per gli sviluppatori Apple.

1. Creazione di un account sviluppatore con Apple.
2. Accedi con le tue credenziali Apple.
3. Sulla barra di navigazione a sinistra, scegli Certificates, Identifiers & Profiles (Certificati, identificatori e profili).
4. Nel riquadro di navigazione a sinistra, scegli Identifiers (Identificatori).
5. Nella pagina Identifiers (Identificatori), scegli l'icona +.
6. Nella pagina Register a New Identifier (Registra un nuovo identificatore), scegli App IDs (ID app), quindi scegli Continue (Continua).

7. Nella pagina Seleziona un tipo, scegli App, quindi scegli Continua.
8. Nella pagina Register an App ID (Registra un'ID app), procedi come indicato di seguito:
    1. In Description (Descrizione), inserisci una descrizione.
    2. In App ID Prefix (Prefisso ID app), inserisci un Bundle ID ID bundle. Prendi nota del valore alla voce Prefisso ID app. Ne avrai bisogno per configurare Apple come provider di identità in Fase 2: aggiunta di un IdP social al bacino d'utenza.
    3. In Funzionalità, scegli Accedi con Apple, quindi scegli Modifica.
    4. Nella pagina Sign in with Apple: App ID Configuration (Accedi con Apple: configurazione ID app) scegli di configurare l'app come app primaria o raggruppata con altri ID app, quindi scegli Save (Salva).
    5. Scegli Continua.
9. Nella pagina Confirm your App ID (Conferma l'ID app), scegli Register (Registra).
10. Nella pagina Identifiers (Identificatori), scegli l'icona +.
11. Nella pagina Register a New Identifier (Registra un nuovo identificatore), seleziona Services IDs (ID servizi), quindi scegli Continue (Continua).
12. Nella pagina Register a Services ID (Registra un ID servizi), procedi nel modo seguente:
    1. In Description (Descrizione), inserisci una descrizione.
    2. Alla voce Identifier (Identificatore), inserisci un identificatore. Prendi nota di questo ID dei servizi perché avrai bisogno di questo valore dopo aver scelto Apple come provider di identità inFase 2: aggiunta di un IdP social al bacino d'utenza.
    3. Scegli Continua, quindi scegli Registra.
13. Scegli l'ID dei servizi che hai appena creato dalla pagina Identificatori.
    1. Scegli Sign In with Apple (Accedi con Apple), quindi scegli Configure (Configura).
    2. Nella pagina Web Authentication Configuration (Configurazione autenticazione Web), seleziona l'ID dell'app creato in precedenza come Primary App ID (ID app principale).
    3. Scegli l'icona + accanto a Website URLs (URL siti Web).
    4. In Domains and subdomains (Domini e sottodomini), inserisci il dominio del pool di utenti senza prefisso `https://`.

        ```
        <your-user-pool-domain>
        ```

5. In Return URLs (URL restituiti), inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

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

6. Scegli Avanti, quindi scegli Fine. Non è necessario verificare il dominio.
7. Scegli Continua, quindi Salva.

14. Nel riquadro di navigazione a sinistra, scegli Keys (Chiavi).
15. Nella pagina Keys (Chiavi), scegli l'icona +.
16. Nella pagina Register a New Key, (Registra una nuova chiave) procedi nel modo seguente:

    1. Alla voce Key Name (nome chiave), digita un nome della chiave.
    2. Scegli Accedi con Apple, quindi scegli Configura.
    3. Nella pagina Configure Key, seleziona l'ID dell'app che hai creato in precedenza come ID app principale. Selezionare Salva.
    4. Scegli Continue (Continua), quindi scegli Register (Registra).
17. Nella pagina Scarica la tua chiave, scegli Scarica per scaricare la chiave privata, prendi nota dell'ID chiave mostrato, quindi scegli Fine. Avrai bisogno di questa chiave privata e del valore di Key ID (ID chiave) visualizzato in questa pagina dopo aver scelto Apple come provider di identità in Fase 2: aggiunta di un IdP social al bacino d'utenza.

Aggiunta di un IdP social al bacino d'utenza

In questa sezione viene configurato un IdP social nel bacino d'utenza utilizzando l'ID client e il segreto client della sezione precedente.

Per configurare un provider di identità social per un pool di utenti con il AWS Management Console

1. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua Federated sign-in (accesso federato) e seleziona Add an identity provider (aggiungi un provider di identità).
5. Scegli un provider di identità social: Facebook, Google, Login with Amazon o Accedi con Apple.
6. Scegli uno tra i seguenti passaggi, in base alla tua scelta del provider di identità social:

- Google e Login with Amazon: inserisci l'ID client dell'app e il segreto del client dell'app generati nella sezione precedente.
- Facebook: inserisci l'ID client dell'app e il segreto del client dell'app generati nella sezione precedente, quindi scegli una versione dell'API (ad esempio, la versione 2.12). Ti consigliamo di scegliere la versione più recente possibile: ogni API di Facebook ha un ciclo di vita e una data di obsolescenza. Gli ambiti e gli attributi di Facebook possono variare a seconda delle versioni dell'API. Ti consigliamo di testare l'accesso alla tua identità social con Facebook per assicurarti che la federazione funzioni come previsto.
- Accedi con Apple: inserisci l'ID dei servizi, l'ID del team, l'ID della chiave e la chiave privata generati nella sezione precedente.

7. Inserisci i nomi degli ambiti autorizzati che desideri utilizzare. Gli ambiti definiscono a quali attributi utente (ad esempio `name` e `email`) intendi accedere con l'app. Per Facebook, devono essere separati da virgole. Per Google e Login with Amazon, devono essere separati da spazi. Per Sign in with Apple, seleziona le caselle di controllo per gli ambiti cui desideri accedere.

Provider di identità social	Ambiti di esempio
Facebook	`public_profile, email`
Google	`profile email openid`
Login with Amazon	`profile postal_code`
Accedi con Apple	`email name`

All'utente dell'app viene richiesto il consenso a fornire questi attributi all'app. Per ulteriori informazioni sugli ambiti dei provider di social, consulta la documentazione di Google, Facebook, Login with Amazon e Accedi con Apple.

Nel caso di Accedi con Apple, di seguito sono riportati scenari utente in cui gli ambiti potrebbero non essere restituiti:

- Un utente finale riscontra degli errori dopo aver lasciato la pagina di accesso di Apple (possono essere dovuti a errori interni di Amazon Cognito o a qualsiasi altro errore scritto dallo sviluppatore).
- L'identificatore ID del servizio viene utilizzato tra pool di utenti e/o altri servizi di autenticazione.

- Uno sviluppatore aggiunge altri ambiti dopo che l'utente ha effettuato l'accesso. Gli utenti recuperano nuove informazioni solo quando si autenticano e quando aggiornano i token.
- Uno sviluppatore elimina l'utente e quindi l'utente accede nuovamente senza rimuovere l'app dal proprio profilo ID Apple.

8. Mappa gli attributi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, consulta Cose da sapere sulle mappature.
9. Scegli Create (Crea) .
10. Dalla scheda App client integration (integrazione client dell'app), scegli uno dei client dell'app nella lista e modifica le impostazioni dell'interfaccia utente ospitata. Aggiungi il nuovo provider di identità social al client dell'app alla voce provider di identità.
11. Scegli Save changes (Salva modifiche).

Test della configurazione dell'IdP social

Puoi creare un URL di accesso utilizzando gli elementi delle due sezioni precedenti. Utilizzalo per testare la tua configurazione IdP social.

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

Per individuare il dominio, vai alla pagina della console Domain name (Nome dominio) del bacino d'utenza. Il valore di client_id è disponibile nella pagina App client settings Impostazioni client di applicazioni. Utilizza l'URL di callback per il parametro redirect_uri. Questo è l'URL della pagina a cui l'utente verrà reindirizzato dopo aver completato la procedura di autenticazione.

> **Note**
>
> Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro 5 minuti e reindirizza l'utente all'interfaccia utente ospitata. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

Aggiunta di un accesso con un provider di identità SAML a un bacino d'utenza (facoltativo)

È possibile abilitare gli utenti della tua app ad accedere tramite un provider di identità (IdP) SAML. Sia che effettuino l'accesso direttamente o attraverso terze parti, tutti gli utenti hanno un profilo nel bacino d'utenza. Salta questa fase se non desideri aggiungere l'accesso attraverso un provider di identità SAML.

Per ulteriori informazioni, consulta Utilizzo di provider di identità SAML con un pool di utenti.

Devi aggiornare il tuo provider di identità SAML e configurare il tuo pool di utenti. Per informazioni su come aggiungere il tuo pool di utenti come relying party o applicazione per il tuo provider di identità SAML 2.0, consulta la documentazione del tuo provider di identità SAML.

È inoltre necessario fornire un endpoint ACS (Assertion Consumer Service) al provider di identità SAML. Configura il seguente endpoint nel dominio del pool di utenti per il binding SAML 2.0 POST nel gestore dell'identità digitale SAML. Per ulteriori informazioni sui domini del pool di utenti, consulta. Configurazione di un dominio di bacino d'utenza

```
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
```

Puoi trovare il prefisso del dominio e il valore della regione per il tuo pool di utenti nella scheda Nome di dominio della console Amazon Cognito.

Per alcuni provider di identità SAML, devi anche fornire il service provider (SP)`urn`, chiamato anche URI del pubblico o ID dell'entità SP, nel formato:

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

Puoi trovare l'ID del bacino d'utenza nella scheda Impostazioni generali nella console Amazon Cognito.

È inoltre necessario configurare il provider di identità SAML per fornire i valori di attributo per tutti gli attributi obbligatori nel bacino d'utenza. Di solito, `email` è un attributo obbligatorio per i bacini

d'utenza. In questo caso, il provider di identità SAML deve fornire un valore `email` (attestazione) nell'asserzione SAML.

I bacini d'utenza di Amazon Cognito supportano la federazione SAML 2.0 con endpoint post-binding. Ciò elimina la necessità per l'app di recuperare o analizzare le risposte alle asserzioni SAML, poiché il pool di utenti riceve direttamente la risposta SAML dal tuo provider di identità tramite uno user agent.

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua Federated sign-in (accesso federato) e seleziona Add an identity provider (aggiungi un provider di identità).
5. Scegli un provider SAML di identità social.
6. Inserisci gli Identifiers (identificatori) separati da virgole. Un identificatore indica ad Amazon Cognito che deve controllare l'indirizzo e-mail che un utente inserisce quando effettua l'accesso. Quindi li indirizza al provider che corrisponde al loro dominio.
7. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. È necessario configurare il provider di identità SAML 2.0 per inviare le risposte di disconnessione all'endpoint `https://<your Amazon Cognito domain>/saml2/logout` creato quando si configura l'interfaccia utente ospitata. L'`saml2/logout`endpoint utilizza l'associazione POST.

   > **Note**
   >
   > Se questa opzione è selezionata e il tuo provider di identità SAML prevede una richiesta di disconnessione firmata, dovrai anche configurare il certificato di firma fornito da Amazon Cognito con il tuo IdP SAML.
   > L'IdP SAML elaborerà la richiesta di disconnessione firmata e disconnetterà l'utente dalla sessione di Amazon Cognito.

8. Scegli una Metadata document source (Fonte del documento di metadati). Se il tuo provider di identità fornisce metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona

Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

> **Note**
>
> Ti consigliamo di inserire l'URL di un documento di metadati se il tuo provider ha un endpoint pubblico, anziché caricare un file. Ciò consente ad Amazon Cognito di aggiornare automaticamente i metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

9. Seleziona Map attributes between your SAML provider and your app (mappare gli attributi tra il provider SAML e la tua app) per mappare gli attributi del provider SAML al profilo utente nel bacino d'utenza. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

   Ad esempio, quando si scegli l'User pool attribute (attributo del bacino d'utenza) `email`, inserisci il nome dell'attributo SAML come compare nell'asserzione SAML dal provider di identità. Il tuo provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni provider di identità utilizzano nomi semplici, come `email`, mentre altri utilizzano attributi con formato URL simili a questo:

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

10. Scegli Create (Crea).

Guida introduttiva ai pool di identità di Amazon Cognito

I bacini d'utenza di Amazon Cognito consentono di creare identità univoche e assegnare le autorizzazioni agli utenti. Il tuo pool di identità può includere:

- Utenti in un bacino d'utenza di Amazon Cognito
- Utenti che eseguono l'autenticazione con un provider di identità esterno come Facebook, Google, Apple o un provider di identità OIDC o SAML.
- Utenti autenticati tramite il tuo processo di autenticazione esistente

Con un pool di identità, puoi ottenere AWS credenziali temporanee con autorizzazioni da te definite per accedere direttamente ad altre risorse Servizi AWS o per accedere tramite Amazon API Gateway.

Argomenti

- Creazione di un pool di identità in Amazon Cognito
- Configurazione di un SDK
- Integrazione dei provider di identità
- Ottenere le credenziali

Creazione di un pool di identità in Amazon Cognito

Puoi creare un bacino d'utenza tramite la console di Amazon Cognito oppure puoi utilizzare l' AWS Command Line Interface (CLI) o le API di Amazon Cognito.

Per creare un nuovo pool di identità nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità.
2. Scegli Crea pool di identità.
3. In Configurazione dell'attendibilità del pool di identità, scegli di configurare il pool di identità per Accesso autenticato, Accesso guest o entrambi.

    - Se hai scelto Accesso autenticato, seleziona uno o più Tipi di identità che desideri impostare come origine delle identità autenticate nel pool di identità. Se configuri un Provider degli sviluppatori personalizzato, non puoi modificarlo né eliminarlo dopo aver creato il pool di identità.

4. In Configura le autorizzazioni, scegli un ruolo IAM predefinito per gli utenti autenticati o guest nel pool di identità.

   a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza il documento di policy per esaminare le autorizzazioni che verranno assegnate da Amazon Cognito al nuovo ruolo IAM.

   b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

5. In Connect identity providers, inserisci i dettagli dei provider di identità (IdPs) che hai scelto in Configure identity pool trust. È possibile che ti venga chiesto di fornire informazioni sul client dell'app OAuth, scegliere un pool di utenti Amazon Cognito, scegliere un IdP IAM o inserire un identificatore personalizzato per un provider degli sviluppatori.

   a. Scegli le impostazioni del ruolo per ogni IdP. Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con attestazione preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

      i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

      ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

   b. Configura Attributi per il controllo degli accessi per ciascun IdP. L'opzione Attributi per il controllo degli accessi associa le richieste dell'utente ai tag principali applicati da Amazon

Cognito alla relativa sessione temporanea. Puoi creare policy IAM per filtrare l'accesso utente in base ai tag applicati alla relativa sessione.

i. Per non applicare alcun tag principale, scegli Inattivo.

ii. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

iii. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

6. In Configura proprietà, inserisci un Nome in Nome del pool di identità.
7. In Autenticazione di base (classica), scegli Attiva flusso di base, se desiderato. Con il flusso di base attivo, puoi ignorare le selezioni di ruolo che hai effettuato per te IdPs e chiamare AssumeRoleWithWebIdentitydirettamente. Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità (identità federate).
8. In Tag, scegli Aggiungi tag se desideri applicare tag al pool di identità.
9. In Esamina e crea, conferma le selezioni effettuate per il nuovo pool di identità. Seleziona Modifica per tornare alla procedura guidata e modificare le eventuali impostazioni. Al termine, seleziona Crea un pool di identità.

Configurazione di un SDK

Per utilizzare i pool di identità di Amazon Cognito, configura AWS Amplify AWS SDK for Java, o il. AWS SDK for .NET Per ulteriori informazioni, consulta i seguenti argomenti.

- Configurazione dell'SDK per JavaScript nella Guida per gli sviluppatori AWS SDK for Java
- Documentazione di Amplify in Amplify Dev Center
- Provider di credenziali Amazon Cognito nella Guida per gli sviluppatori di AWS SDK for .NET

Integrazione dei provider di identità

In pool di identità di Amazon Cognito (identità federate) supportano l'autenticazione degli utenti tramite i pool di utenti Amazon Cognito, i provider di identità federate (inclusi i provider di identità Amazon, Facebook, Google e SAML) e le identità non autenticate. Questa funzione supporta anche Identità autenticate dagli sviluppatori (pool di identità), che ti consente di registrare e di autenticare gli utenti tramite il tuo processo di autenticazione di back-end.

Per ulteriori informazioni sull'utilizzo di un bacino d'utenza di Amazon Cognito per la creazione della tua directory utente, consulta Bacini d'utenza di Amazon Cognito e Accesso Servizi AWS tramite un pool di identità dopo l'accesso.

Per ulteriori informazioni su come utilizzare provider di identità esterni, consulta Provider di identità esterni con pool di identità.

Per ulteriori informazioni su come integrare il tuo processo di autenticazione di back-end, consulta Identità autenticate dagli sviluppatori (pool di identità).

Ottenere le credenziali

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per gli utenti ospiti (non autenticati) e per gli utenti che si sono autenticati e hanno ricevuto un token. Con queste AWS credenziali, la tua app può accedere in modo sicuro a un backend interno AWS o esterno AWS tramite Amazon API Gateway. Per informazioni, consulta Ottenere le credenziali.

Opzioni di configurazione guidate per Amazon Cognito

Potresti voler valutare le funzionalità di Amazon Cognito in un'esperienza strutturata e guidata. Ecco alcune risorse esterne che forniscono esperienze personalizzate con pool di utenti e pool di identità.

Completa un workshop

AWS workshop studio ospita un workshop che illustra la configurazione della maggior parte delle funzionalità di Amazon Cognito. Queste funzionalità includono l'API dei pool di utenti, l'interfaccia utente ospitata dai pool di utenti, i pool di identità e la configurazione di sicurezza.

Aggiungi codice applicativo tratto da esempi

Il capitolo sugli esempi di codice di questa guida contiene codice applicativo che è possibile utilizzare con pool di utenti e pool di identità. La sezione pool di utenti del capitolo sugli esempi di codice contiene brevi frammenti che coprono singole operazioni ed esempi più lunghi, end-to-end ad esempio applicazioni in una varietà di linguaggi di programmazione.

Crea un'applicazione fullstack con AWS Amplify

AWS Amplifyè Servizio AWS destinato agli sviluppatori che desiderano sviluppare e ospitare un'applicazione e un'interfaccia utente. Amazon Cognito è il componente di autenticazione di Amplify. Quando aggiungi l'autenticazione alla tua applicazione, Amplify può automatizzare la distribuzione del pool di utenti e delle risorse del pool di identità di Amazon Cognito. Consulta anche Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili.

Altre risorse sulle applicazioni Amazon Cognito su GitHub

- Esempi di flussi di autenticazione con.NET per Amazon Cognito
- Autenticazione senza password di Amazon Cognito
- PetStoreesempio con Amazon Verified Permissions
- Esempio di app React che utilizza ABAC + Identity Pools per accedere alle risorse AWS
- Autorizzazione da macchina a macchina basata su Amazon Cognito e API Gateway tramite CDK AWS
- Creazione di autorizzazioni granulari utilizzando Amazon Cognito, API Gateway e IAM
- CloudFrontautorizzazione @edge

Altri workshop

- Implementa l'autenticazione senza password con Amazon Cognito e WebAuthn
- Identità SaaS multi-tenant con pool di utenti Amazon Cognito
- Approfondimento su Amazon Cognito JWT

Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili

Quando integri la tua app con un client di app Amazon Cognito, puoi richiamare le operazioni API per l'autenticazione e l'autorizzazione dei tuoi utenti. Ti consigliamo di utilizzarlo AWS Amplifyper integrare Amazon Cognito con le tue app web e mobili. AWS Amplify è una soluzione completa che consente agli sviluppatori web e mobili di frontend di creare, connettere e ospitare facilmente applicazioni full stack AWS, con la flessibilità necessaria per sfruttare l'ampiezza dei casi d'uso man mano che i casi d'uso si evolvono. Servizi AWS Amplify Auth utilizza principalmente Amazon Cognito per creare funzionalità di autenticazione.

Argomenti

- Autenticazione con AWS Amplify
- Autenticazione con SDK AWS
- Autorizzazione con Amazon Verified Permissions

Un'implementazione tipica di Amazon Cognito utilizza una combinazione di strumenti visivi e API. La console Amazon Cognito è l'interfaccia visiva per la configurazione e la gestione dei pool di utenti e dei pool di identità di Amazon Cognito. L'interfaccia utente ospitata è un'applicazione di accesso ready-to-use basata sul Web per il test e la distribuzione rapidi dei pool di utenti di Amazon Cognito. Inoltre, nella maggior parte delle implementazioni di Amazon Cognito è necessario aggiungere codice nelle app per interagire con i pool di utenti e i pool di identità. Ad esempio, l'app potrebbe richiamare l'interfaccia utente ospitata per l'accesso utente, quindi chiamare l'endpoint del token dal codice dell'app per scambiare il codice di autorizzazione dell'utente con i token. Quindi, l'app deve interpretare e archiviare i token dell'utente e presentarli nel contesto appropriato per l'autenticazione e l'autorizzazione. Amplify aggiunge strumenti di integrazione guidata con funzioni integrate per tali processi.

Puoi anche creare le risorse Amazon Cognito interamente nel codice. Per iniziare a usare il codice dell'app, visita gli esempi di codice di Amazon Cognito per gli SDK AWS. Per l'integrazione con Amazon Cognito come provider di identità OpenID Connect, usa Strumenti per sviluppatori OpenID Connect.

Prima di utilizzare l'autenticazione e l'autorizzazione Amazon Cognito, scegli una piattaforma app e prepara il tuo codice per l'integrazione con il servizio. Per le piattaforme disponibili,

consulta Autenticazione con SDK AWS. AWS CLI È un SDK a riga di comando per Amazon Cognito e altri Servizi AWS ed è un ottimo punto di partenza per iniziare a familiarizzare con l'API Amazon Cognito.

> **Note**
>
> Puoi configurare alcuni componenti di Amazon Cognito solo con l'API. Ad esempio, puoi impostare un trigger Lambda personalizzato per SMS o mittente e-mail del pool di utenti solo con una richiesta che aggiorna `LambdaConfig` la proprietà UserPooldella classe in CreateUserPooluna richiesta UpdateUserPoolo API.

L'API dei pool di utenti di Amazon Cognito condivide lo spazio dei nomi con diverse classi di operazioni API. Una classe configura i pool di utenti e i relativi processi, i provider di identità e gli utenti. Un'altra include operazioni non autenticate che consentono agli utenti di un client pubblico di accedere, disconnettersi e gestire i propri profili. L'ultima classe di operazioni API esegue operazioni utente autorizzate con le proprie AWS credenziali in un client riservato lato server. È necessario conoscere l'architettura dell'app desiderata prima di iniziare a implementare il codice dell'app. Per ulteriori informazioni, consulta Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti.

Autenticazione con AWS Amplify

AWS Amplify è una soluzione completa per la creazione di applicazioni web e mobili. Con Amplify, puoi connetterti alle risorse esistenti con le librerie Amplify oppure puoi creare e configurare nuove risorse con l'interfaccia a riga di comando (CLI) di Amplify. Amplify dispone anche di componenti dell'interfaccia utente collegati come Autenticatore per configurare e personalizzare l'esperienza di accesso e registrazione nella tua app.

Per utilizzare le funzionalità di autenticazione di Amplify nella tua app front-end, consulta la seguente documentazione in base alla piattaforma.

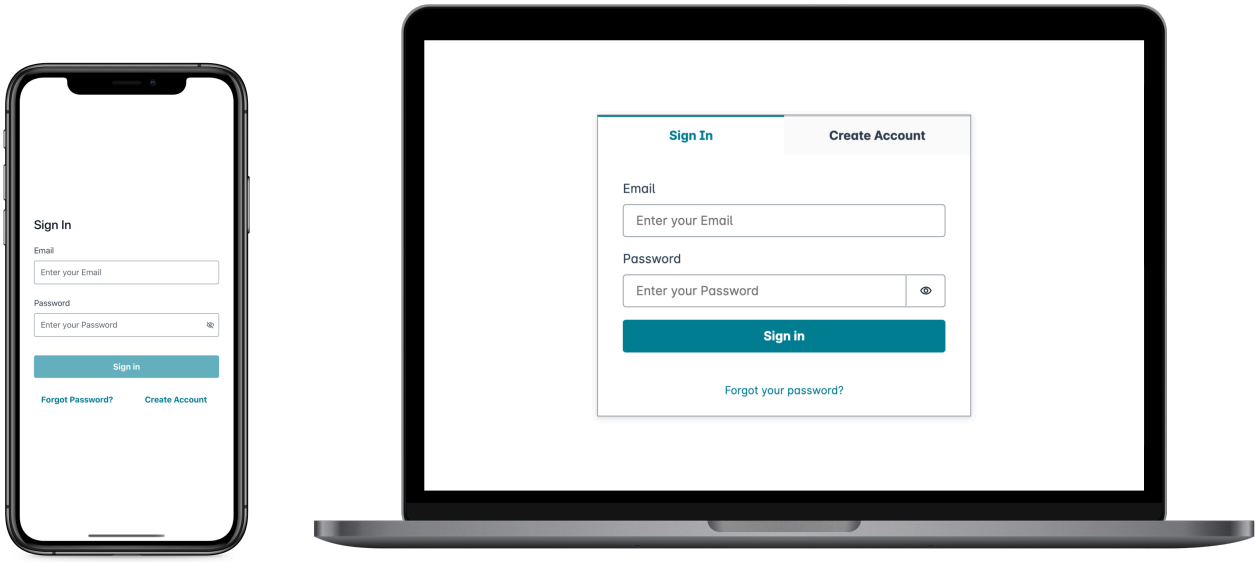
- Autenticazione Amplify per JavaScript
- Autenticazione Amplify per iOS
- Autenticazione Amplify per Android
- Amplifica l'autenticazione per Flutter

Le librerie Amplify sono open source e sono disponibili su. GitHub Per ulteriori informazioni su come Amplify Auth implementa l'autenticazione Amazon Cognito, visita le seguenti librerie.

- amplifica-js
- amplify-swift
- amplify-flutter
- amplify-android

Creazione di un'interfaccia utente con Amplify

L'interfaccia utente ospitata dei pool di utenti di Amazon Cognito può soddisfare le esigenze essenziali di un front-end di autenticazione per un'app Web o per dispositivi mobili. Per personalizzare l'interfaccia utente oltre i parametri consentiti dall'interfaccia utente ospitata, crea un'app personalizzata. Amplify UI è una raccolta personalizzabile di componenti front-end in un'ampia gamma di lingue.



Per iniziare a utilizzate il tuo componente di autenticazione personalizzato, consulta la seguente documentazione per il componente Authenticator.

- Authenticator per Android
- Authenticator per Angular

- Authenticator per Angular
- Authenticator per React
- Authenticator per React Native
- Authenticator per Swift
- Authenticator per Vue

Autenticazione con SDK AWS

Per utilizzare un backend sicuro per creare il tuo microservizio di identità che interagisce con Amazon Cognito, connettiti ai pool di utenti di Amazon Cognito e all'API dei pool di identità di Amazon Cognito con AWS un SDK nella lingua che preferisci.

Per informazioni dettagliate su tutte le operazioni API, consulta la Documentazione di riferimento dell'API di Amazon Cognito e la Documentazione di riferimento dell'API Amazon Cognito. Questi documenti contengono sezioni vedi anche, che includono risorse per l'utilizzo di una varietà di SDK nelle piattaforme supportate.

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK for Go
- AWS SDK per Java V2
- AWS SDK per JavaScript
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

Autorizzazione con Amazon Verified Permissions

Amazon Verified Permissions è un servizio di autorizzazione per le applicazioni che crei. Quando aggiungi un pool di utenti Amazon Cognito come fonte di identità, la tua app può passare i token di accesso o di identità (ID) al pool di utenti alle Autorizzazioni verificate per consentire o negare una decisione. Verified Permissions considera le proprietà dell'utente e il contesto della richiesta

in base alle politiche redatte in Cedar Policy Language. Il contesto della richiesta può includere un identificatore per il documento, l'immagine o l'altra risorsa richiesta e l'azione che l'utente desidera intraprendere sulla risorsa.

L'app può fornire l'identità dell'utente o i token di accesso alle autorizzazioni verificate nelle richieste API. IsAuthorizedWithTokenBatchIsAuthorizedWithToken Queste operazioni API accettano gli utenti come utenti `Principal` e prendono decisioni di `Action` autorizzazione per `Resource` chi desidera accedere. Ulteriori personalizzazioni `Context` possono contribuire a una decisione di accesso dettagliata.

Quando la tua app presenta un token in una richiesta `IsAuthorizedWithToken` API, Verified Permissions esegue le seguenti convalide.

1. Il tuo pool di utenti è un'origine di identità di Verified Permissions configurata per il policy store richiesto.
2. La richiesta `client_id` o `aud`, rispettivamente, nel tuo token di accesso o di identità, corrisponde all'ID client dell'app pool di utenti che hai fornito a Verified Permissions. Per verificare questa richiesta, devi configurare la convalida dell'ID cliente nella tua fonte di identità Autorizzazioni verificate.
3. Il tuo token non è scaduto.
4. Il valore dell'`token_use`attestazione nel tuo token corrisponde ai parametri a cui lo hai passato`IsAuthorizedWithToken`. L'`token_use`attestazione deve essere `access` se l'hai passata al `accessToken` parametro e `id` se l'hai passata al `identityToken` parametro.
5. La firma nel tuo token proviene dalle chiavi web JSON pubblicate (JWK) del tuo pool di utenti. Puoi visualizzare i tuoi JWK su `https://cognito-idp.`*`Region`*`.amazonaws.com/`*`your user pool ID`*`/.well-known/jwks.json`.

Token revocati e utenti eliminati

Verified Permissions convalida solo le informazioni che conosce dalla fonte della tua identità e dalla data di scadenza del token dell'utente. Verified Permissions non verifica la revoca del token o l'esistenza dell'utente. Se hai revocato il token dell'utente o hai eliminato il profilo utente dal tuo pool di utenti, Verified Permissions considera comunque il token valido fino alla scadenza.

Valutazione delle politiche

Configura il tuo pool di utenti come fonte di identità per il tuo policy store. Configura la tua app per inviare i token degli utenti nelle richieste a Verified Permissions. Per ogni richiesta, Verified

Permissions confronta le affermazioni contenute nel token con una politica. Una politica di Verified Permissions è come una politica IAM in AWS. Dichiara un principio, una risorsa e un'azione. Verified Permissions risponde alla tua richiesta indicando `Allow` se corrisponde a un'azione consentita e non corrisponde a un'`Deny`azione esplicita; in caso contrario, risponde con. `Deny` Per ulteriori informazioni, consulta le politiche relative a Amazon Verified Permissions nella Guida per l'utente di Amazon Verified Permissions.

Personalizzazione dei token

Per modificare, aggiungere e rimuovere le affermazioni degli utenti che desideri presentare a Verified Permissions, personalizza il contenuto dei tuoi token di accesso e di identità con un. Trigger Lambda di pre-generazione del token Con un trigger prima della generazione di token, puoi aggiungere e modificare le richieste nei tuoi token. Ad esempio, puoi interrogare un database per gli attributi utente aggiuntivi e codificarli nel tuo token ID.

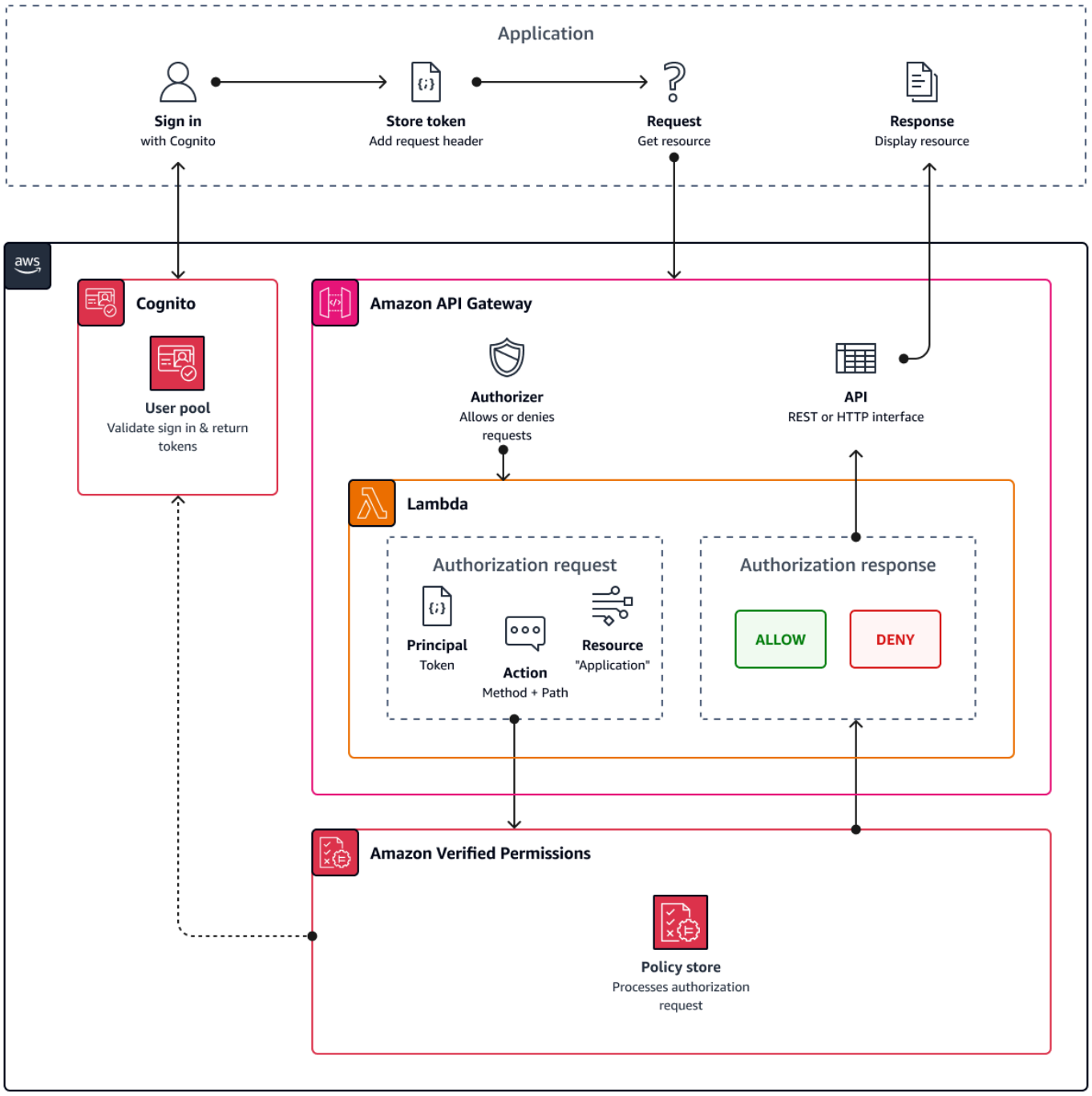
> **Note**
>
> A causa del modo in cui Verified Permissions elabora le richieste, non aggiungerle con nome `cognito` , `dev` o `custom` nella funzione di pre-generazione del token. Quando presenti questi prefissi riservati per richieste non in un formato delimitato da due punti, ad esempio come `cognito:username`, ma come nomi completi delle richieste, le tue richieste di autorizzazione hanno esito negativo.

Per ulteriori informazioni su come Autorizzazioni verificate associa attestazioni nei token Amazon Cognito alle policy di autorizzazione, consulta Mapping Amazon Cognito tokens to Verified Permissions schema.

Autorizzazione API con autorizzazioni verificate

Il tuo ID o i token di accesso possono autorizzare le richieste alle API REST di Amazon API Gateway di back-end con autorizzazioni verificate. Puoi creare un archivio di policy con collegamenti immediati al tuo pool di utenti e alla tua API. Con l'opzione di avvio Configurazione con Cognito e API Gateway, Verified Permissions aggiunge una fonte di identità del pool di utenti al policy store e un autorizzatore Lambda all'API. Quando l'applicazione passa un token portatore del pool di utenti all'API, l'autorizzatore Lambda richiama le autorizzazioni verificate. L'autorizzatore passa il token come principale e il percorso e il metodo della richiesta come azione.

Il diagramma seguente illustra il flusso di autorizzazione per un'API API Gateway con autorizzazioni verificate. Per un'analisi dettagliata, consulta gli archivi di policy collegati alle API nella Amazon Verified Permissions User Guide.



Verified Permissions struttura l'autorizzazione delle API in base ai gruppi di pool di utenti. Poiché sia l'ID che i token di accesso includono un `cognito:groups` claim, il policy store può gestire il controllo degli accessi basato sui ruoli (RBAC) per le API in una varietà di contesti applicativi.

Scelta delle impostazioni del policy store

Quando si configura un'origine di identità in un policy store, è necessario scegliere se elaborare i token di accesso o ID. Questa decisione è importante per il modo in cui funziona il motore delle policy. I token ID contengono attributi utente. I token di accesso contengono informazioni sul controllo degli accessi degli utenti: ambiti OAuth. Sebbene entrambi i tipi di token contengano informazioni sull'appartenenza al gruppo, in genere consigliamo il token di accesso per RBAC con un archivio di policy di autorizzazioni verificate. Il token di accesso si aggiunge all'appartenenza al gruppo con ambiti che possono contribuire alla decisione di autorizzazione. Le affermazioni in un token di accesso diventano contesto nella richiesta di autorizzazione.

È inoltre necessario configurare i tipi di entità utente e di gruppo quando si configura un pool di utenti come fonte di identità. I tipi di entità sono identificatori principali, di azioni e di risorse a cui puoi fare riferimento nelle politiche di autorizzazione verificate. Le entità negli archivi delle politiche possono avere una relazione di appartenenza, in cui un'entità può essere membro di un'entità principale. Con l'appartenenza, puoi fare riferimento a gruppi principali, gruppi di azione e gruppi di risorse. Nel caso di gruppi di pool di utenti, il tipo di entità utente specificato deve essere un membro del tipo di entità di gruppo. Quando configuri un policy store collegato all'API o segui la configurazione guidata nella console Verified Permissions, il policy store ha automaticamente questa relazione genitore-membro.

Il token ID può combinare RBAC con il controllo degli accessi basato sugli attributi (ABAC). Dopo aver creato un policy store collegato all'API, puoi migliorare le tue policy con gli attributi utente e l'appartenenza ai gruppi. Le dichiarazioni di attributo in un token ID diventano gli attributi principali nella richiesta di autorizzazione. Le tue politiche possono prendere decisioni di autorizzazione in base agli attributi principali.

Puoi anche configurare un policy store per accettare token con un'`client_id`affermazione `aud` or che corrisponda a un elenco di app client accettabili da te fornito.

Esempio di politica per l'autorizzazione delle API basata sui ruoli

La seguente politica di esempio è stata creata configurando un archivio di criteri di autorizzazione verificata per un'API REST di PetStoreesempio.

```
permit(
```

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

Verified Permissions restituisce una `Allow` decisione sulla richiesta di autorizzazione dell'applicazione quando:

1. L'applicazione ha passato un ID o un token di accesso in un'`Authorization`intestazione come token portatore.
2. L'applicazione ha passato un token con un'`cognito:groups`affermazione che contiene la stringa. `MyGroup`
3. L'applicazione ha inviato una `HTTP GET` richiesta, ad esempio, a `https://myapi.example.com/pets` o`https://myapi.example.com/pets/scrappy`.

Esempio di policy per un utente Amazon Cognito.

Il tuo pool di utenti può anche generare richieste di autorizzazione a Verified Permissions in condizioni diverse dalle richieste API. Puoi inviare qualsiasi decisione di controllo degli accessi contenuta nella tua applicazione al tuo policy store. Ad esempio, puoi integrare la sicurezza di Amazon DynamoDB o Amazon S3 con il controllo degli accessi basato sugli attributi prima che le richieste transitino sulla rete, riducendo l'utilizzo delle quote.

L'esempio seguente utilizza il Cedar Policy Language per consentire agli utenti Finance che si autenticano con un client dell'app pool di utenti di leggere e scrivere `example_image.png`. John, un utente della tua app, riceve un token ID dal client dell'app e lo trasmette in una richiesta GET a un URL che richiede l'autorizzazione, `https://example.com/images/example_image.png`. Il token ID di John ha una richiesta `aud` dell'ID client dell'app del pool di utenti `1234567890example`. La tua funzione Lambda di prima generazione del token ha anche inserito una nuova affermazione `costCenter` con un valore, per John, di `Finance1234`.

```
permit (
   principal,
   actions in [ExampleCorp::Action::"readFile", "writeFile"],
   resource == ExampleCorp::Photo::"example_image.png"
)
when {
   principal.aud == "1234567890example" &&
```

```
   principal.custom.costCenter like "Finance*"
};
```

Il seguente corpo di richiesta restituisce una risposta `Allow`.

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

Quando desideri specificare un principale in una politica di autorizzazioni verificate, utilizza il seguente formato:

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

Di seguito è riportato un esempio principale per un utente in un pool di utenti con ID `us-east-1_Example` con ID secondario o ID utente. `973db890-092c-49e4-a9d0-912a4c0a20c7`

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

Quando desideri specificare un gruppo di utenti in una politica di autorizzazioni verificate, utilizza il seguente formato:

```
permit (
   principal in [Namespace]::[Group Entity]::"[Group name]",
   action,
   resource
);
```

Di seguito è riportato un esempio

Controllo dell'accesso basato sugli attributi

L'autorizzazione con autorizzazioni verificate per le tue app e gli attributi per la funzionalità di controllo degli accessi dei pool di identità di Amazon Cognito AWS per le credenziali sono entrambe forme di controllo degli accessi basato sugli attributi (ABAC). Di seguito è riportato un confronto tra le funzionalità di Verified Permissions e Amazon Cognito ABAC. In ABAC, un sistema esamina gli attributi di un'entità e prende una decisione di autorizzazione in base a condizioni definite dall'utente.

Servizio	Processo	Risultato
Autorizzazioni verificate da Amazon	Restituisce una `Deny` decisione `Allow` or dall'analisi di un pool di utenti JWT.	L'accesso alle risorse dell'applicazione riesce o fallisce in base alla valutazione delle politiche Cedar.
Pool di identità di Amazon Cognito (attributi per il controllo degli accessi)	Assegna i tag di sessione all'utente in base ai suoi attributi. Le condizioni delle policy IAM possono controllare i tag `Allow` o `Deny` l'accesso degli utenti a Servizi AWS.	Una sessione con tag con AWS credenziali temporanee per un ruolo IAM.

Esempi di codice per Amazon Cognito utilizzando SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare Amazon Cognito con un Software Development Kit (SDK) AWS.

Per un elenco completo delle guide per gli sviluppatori di SDK AWS ed esempi di codice, consulta la sezione Utilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di codice per Amazon Cognito Identity tramite SDK AWS
  - Azioni per Amazon Cognito Identity tramite SDK AWS
    - Utilizzo CreateIdentityPool con un AWS SDK o una CLI
    - Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
    - Utilizzo DescribeIdentityPool con un AWS SDK o una CLI
    - Utilizzo GetCredentialsForIdentity con un AWS SDK o una CLI
    - Utilizzo GetIdentityPoolRoles con un AWS SDK o una CLI
    - Utilizzo ListIdentityPools con un AWS SDK o una CLI
    - Utilizzo SetIdentityPoolRoles con un AWS SDK o una CLI
    - Utilizzo UpdateIdentityPool con un AWS SDK o una CLI
  - Esempi multiservizio per Amazon Cognito Identity utilizzando SDK AWS
    - Creazione di un'app Amazon Transcribe
    - Creazione di un'applicazione Amazon Textract explorer
- Esempi di codice per Amazon Cognito Identity Provider che utilizza SDK AWS
  - Azioni per Amazon Cognito Identity Provider tramite SDK AWS
    - Utilizzo AdminCreateUser con un AWS SDK o una CLI
    - Utilizzo AdminGetUser con un AWS SDK o una CLI
    - Utilizzo AdminInitiateAuth con un AWS SDK o una CLI
    - Utilizzo AdminRespondToAuthChallenge con un AWS SDK o una CLI
    - Utilizzo AdminSetUserPassword con un AWS SDK o una CLI
    - Utilizzo AssociateSoftwareToken con un AWS SDK o una CLI
    - Utilizzo ConfirmDevice con un AWS SDK o una CLI

        - Utilizzo ConfirmForgotPassword con un AWS SDK o una CLI
        - Utilizzo ConfirmSignUp con un AWS SDK o una CLI
        - Utilizzo CreateUserPool con un AWS SDK o una CLI
        - Utilizzo CreateUserPoolClient con un AWS SDK o una CLI
        - Utilizzo DeleteUser con un AWS SDK o una CLI
        - Utilizzo ForgotPassword con un AWS SDK o una CLI
        - Utilizzo InitiateAuth con un AWS SDK o una CLI
        - Utilizzo ListUserPools con un AWS SDK o una CLI
        - Utilizzo ListUsers con un AWS SDK o una CLI
        - Utilizzo ResendConfirmationCode con un AWS SDK o una CLI
        - Utilizzo RespondToAuthChallenge con un AWS SDK o una CLI
        - Utilizzo SignUp con un AWS SDK o una CLI
        - Utilizzo UpdateUserPool con un AWS SDK o una CLI
        - Utilizzo VerifySoftwareToken con un AWS SDK o una CLI
    - Scenari per Amazon Cognito Identity Provider che utilizzano SDK AWS
        - Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un SDK AWS
        - Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un SDK AWS
        - Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un SDK AWS
        - Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK AWS
- Esempi di codice per Amazon Cognito Sync tramite SDK AWS
    - Azioni per Amazon Cognito Sync tramite SDK AWS
        - Utilizzo ListIdentityPoolUsage con un AWS SDK o una CLI

Esempi di codice per Amazon Cognito Identity tramite SDK AWS

I seguenti esempi di codice mostrano come usare Amazon Cognito Identity con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Azioni per Amazon Cognito Identity tramite SDK AWS
  - Utilizzo CreateIdentityPool con un AWS SDK o una CLI
  - Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
  - Utilizzo DescribeIdentityPool con un AWS SDK o una CLI
  - Utilizzo GetCredentialsForIdentity con un AWS SDK o una CLI
  - Utilizzo GetIdentityPoolRoles con un AWS SDK o una CLI
  - Utilizzo ListIdentityPools con un AWS SDK o una CLI
  - Utilizzo SetIdentityPoolRoles con un AWS SDK o una CLI
  - Utilizzo UpdateIdentityPool con un AWS SDK o una CLI
- Esempi multiservizio per Amazon Cognito Identity utilizzando SDK AWS
  - Creazione di un'app Amazon Transcribe
  - Creazione di un'applicazione Amazon Textract explorer

Azioni per Amazon Cognito Identity tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Identity con AWS gli SDK. Questi estratti chiamano l'API Amazon Cognito Identity e sono estratti di codice da programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per l'elenco completo, consulta Amazon Cognito Identity API Reference (Documentazione di riferimento delle API di identità di Amazon Cognito).

Esempi

- Utilizzo CreateIdentityPool con un AWS SDK o una CLI
- Utilizzo DeleteIdentityPool con un AWS SDK o una CLI
- Utilizzo DescribeIdentityPool con un AWS SDK o una CLI
- Utilizzo GetCredentialsForIdentity con un AWS SDK o una CLI
- Utilizzo GetIdentityPoolRoles con un AWS SDK o una CLI
- Utilizzo ListIdentityPools con un AWS SDK o una CLI
- Utilizzo SetIdentityPoolRoles con un AWS SDK o una CLI
- Utilizzo UpdateIdentityPool con un AWS SDK o una CLI

Utilizzo **`CreateIdentityPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateIdentityPool`.

CLI

AWS CLI

Per creare un pool di identità con il provider del pool di identità Cognito

Questo esempio crea un pool di identità denominato MyIdentityPool. Dispone di un provider del pool di identità Cognito. Le identità non autenticate non sono consentite.

Comando:

```
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_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Output:

```
{
  "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_111111111",
```

```
            "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
            "ServerSideTokenCheck": false
        }
    ]
}
```

- Per i dettagli sull'API, vedere CreateIdentityPoolin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.CognitoIdentityProviderExc

/**
 * 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 "";
    }
}
```

- Per i dettagli sull'API, consulta la CreateIdentityPoolsezione AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: crea un nuovo pool di identità che consente identità non autenticate.

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

Output:

```
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
```

- Per i dettagli sull'API, vedere CreateIdentityPoolin AWS Tools for PowerShell Cmdlet Reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea un nuovo pool di identità.

```
    /// 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
    }
```

- Per ulteriori informazioni, consulta AWS SDK for Swift developer guide (Guida per gli sviluppatori di AWS SDK per Swift).
- Per i dettagli sull'API, consulta la CreateIdentityPoolguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteIdentityPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteIdentityPool`.

CLI

AWS CLI

Per eliminare un pool di identità

Nell'esempio `delete-identity-pool` seguente viene eliminato il pool di identità specificato.

Comando:

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

Questo comando non produce alcun output.

- Per i dettagli sull'API, consulta DeleteIdentityPool AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 identityPoold = args[0];
        CognitoIdentityClient cognitoIdClient = CognitoIdentityClient.builder()
                .region(Region.US_EAST_1)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        deleteIdPool(cognitoIdClient, identityPoold);
        cognitoIdClient.close();
    }

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

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

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

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

```
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteIdentityPoolsezione AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: elimina un pool di identità specifico.

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

- Per i dettagli sull'API, vedere DeleteIdentityPoolin AWS Tools for PowerShell Cmdlet Reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elimina il pool di identità specificato.

```
    /// 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)
    }
```

- Per ulteriori informazioni, consulta AWS SDK for Swift developer guide (Guida per gli sviluppatori di AWS SDK per Swift).
- Per i dettagli sull'API, consulta la DeleteIdentityPoolguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeIdentityPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeIdentityPool`.

CLI

AWS CLI

Per descrivere un pool di identità

Questo esempio descrive un pool di identità.

Comando:

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

Output:

```
{
  "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_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Per i dettagli sull'API, consulta DescribeIdentityPool AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: recupera le informazioni su uno specifico pool di identità in base al relativo id.

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

Output:

```
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
```

- Per i dettagli sull'API, vedere DescribeIdentityPoolin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetCredentialsForIdentity`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`GetCredentialsForIdentity`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.CognitoIdentityProviderExc

/**
 * 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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la GetCredentialsForIdentitysezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetIdentityPoolRoles`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetIdentityPoolRoles`.

CLI

AWS CLI

Per ottenere i ruoli del pool di identità

Questo esempio ottiene i ruoli del pool di identità.

Comando:

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

Output:

```
{
  "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"
  }
}
```

- Per i dettagli sull'API, consulta GetIdentityPoolRoles AWS CLICommand Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: ottiene le informazioni sui ruoli per uno specifico pool di identità.

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

Output:

```
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
```

- Per i dettagli sull'API, vedere GetIdentityPoolRolesin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListIdentityPools`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListIdentityPools`.

CLI

AWS CLI

Per elencare pool di identità

In questo esempio vengono elencati i pool di identità. Sono elencate un massimo di 20 identità.

Comando:

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

Output:

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

- Per i dettagli sull'API, consulta ListIdentityPools AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.CognitoIdentityProviderExc
```

```
/**
 * 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);
        }
    }
}
```

- Per i dettagli sull'API, ListIdentityPoolsconsulta AWS SDK for Java 2.x API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: recupera un elenco di pool di identità esistenti.

```
Get-CGIIdentityPoolList
```

Output:

```
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
```

- Per i dettagli sull'API, vedere ListIdentityPoolsin AWS Tools for PowerShell Cmdlet Reference.

Swift

SDK per Swift

> **Note**
>
> Si tratta di una documentazione di pre-rilascio di un SDK nella versione di anteprima. ed è soggetta a modifiche.

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Trova l'ID di un pool di identità specificandone il nome.

```
    /// 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
```

```
    }
```

Ottieni l'ID di un pool di identità esistente o crealo se non esiste.

```
/// 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
}
```

- Per ulteriori informazioni, consulta AWS SDK for Swift developer guide (Guida per gli sviluppatori di AWS SDK per Swift).
- Per i dettagli sull'API, consulta la ListIdentityPoolsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`SetIdentityPoolRoles`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`SetIdentityPoolRoles`.

CLI

AWS CLI

Per impostare i ruoli del pool di identità

L'`set-identity-pool-roles`esempio seguente imposta un ruolo del pool di identità.

```
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"
```

- Per i dettagli sull'API, vedere SetIdentityPoolRolesin AWS CLI Command Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: configura lo specifico pool di identità per avere un ruolo IAM non autenticato.

```
Set-CGIIdentityPoolRole -IdentityPoolId us-east-1:0de2af35-2988-4d0b-b22d-
EXAMPLEGUID1 -Role @{ "unauthenticated" = "arn:aws:iam::123456789012:role/
CommonTests1Role" }
```

- Per i dettagli sull'API, vedere SetIdentityPoolRolesin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateIdentityPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateIdentityPool`.

CLI

AWS CLI

Per aggiornare un pool di identità

Questo esempio aggiorna un pool di identità. Imposta il nome su MyIdentityPool. Aggiunge Cognito come provider di identità. Non consente identità non autenticate.

Comando:

```
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_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Output:

```
{
  "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_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- Per i dettagli sull'API, consulta UpdateIdentityPoolCommand Reference.AWS CLI

PowerShell

Strumenti per PowerShell

Esempio 1: aggiorna alcune proprietà del pool di identità, in questo caso il nome del pool di identità.

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

Output:

```
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
```

- Per i dettagli sull'API, vedere UpdateIdentityPoolin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi multiservizio per Amazon Cognito Identity utilizzando SDK AWS

Le seguenti applicazioni di esempio utilizzano AWS gli SDK per combinare Amazon Cognito Identity con altri. Servizi AWS Ogni esempio include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire l'applicazione.

Esempi

- Creazione di un'app Amazon Transcribe
- Creazione di un'applicazione Amazon Textract explorer

Creazione di un'app Amazon Transcribe

L'esempio di codice seguente mostra come utilizzare Amazon Transcribe per trascrivere e visualizzare le registrazioni vocali nel browser.

JavaScript

SDK per JavaScript (v3)

Crea un'app che utilizza Amazon Transcribe per trascrivere e visualizzare le registrazioni vocali nel browser. L'app utilizza due bucket Amazon Simple Storage Service (Amazon S3),

uno per ospitare il codice dell'applicazione e l'altro per archiviare le trascrizioni. L'app utilizza un pool di utenti Amazon Cognito per autenticare gli utenti. Gli utenti autenticati dispongono delle autorizzazioni AWS Identity and Access Management (IAM) per accedere ai servizi richiesti. AWS

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Questo esempio è anche disponibile nella Guida per lo sviluppatore di AWS SDK for JavaScript v3 .

Servizi utilizzati in questo esempio

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione di un'applicazione Amazon Textract explorer

Gli esempi di codice seguenti mostrano come esplorare l'output di Amazon Textract tramite un'applicazione interattiva.

JavaScript

SDK per JavaScript (v3)

Mostra come utilizzare per AWS SDK for JavaScript creare un'applicazione React che utilizza Amazon Textract per estrarre dati dall'immagine di un documento e visualizzarli in una pagina Web interattiva. Questo esempio viene eseguito in un browser Web e richiede, come credenziali, un'identità autenticata Amazon Cognito. Utilizza Amazon Simple Storage Service (Amazon S3) per l'archiviazione e per le notifiche esegue il polling di una coda di Servizio di coda semplice Amazon (Amazon SQS) sottoscritta a un argomento Servizio di notifica semplice Amazon (Amazon SNS).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

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

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice per Amazon Cognito Identity Provider che utilizza SDK AWS

I seguenti esempi di codice mostrano come utilizzare Amazon Cognito Identity Provider con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Ciao Amazon Cognito

Gli esempi di codice seguente mostrano come iniziare a utilizzare Amazon Cognito.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per il file CMake C MakeLists .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})
```

Codice per il file origine 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;
```

```
}
```

- Per i dettagli sull'API, consulta API ListUserPoolsReference AWS SDK for C++ .

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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%v: %v\n", *pool.Name, *pool.Id)
  }
 }
}
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
```

```
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);
        }
    }
```

```
}
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
};
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for JavaScript API Reference.

Esempi di codice

- Azioni per Amazon Cognito Identity Provider tramite SDK AWS
  - Utilizzo AdminCreateUser con un AWS SDK o una CLI
  - Utilizzo AdminGetUser con un AWS SDK o una CLI
  - Utilizzo AdminInitiateAuth con un AWS SDK o una CLI
  - Utilizzo AdminRespondToAuthChallenge con un AWS SDK o una CLI
  - Utilizzo AdminSetUserPassword con un AWS SDK o una CLI
  - Utilizzo AssociateSoftwareToken con un AWS SDK o una CLI
  - Utilizzo ConfirmDevice con un AWS SDK o una CLI
  - Utilizzo ConfirmForgotPassword con un AWS SDK o una CLI
  - Utilizzo ConfirmSignUp con un AWS SDK o una CLI
  - Utilizzo CreateUserPool con un AWS SDK o una CLI
  - Utilizzo CreateUserPoolClient con un AWS SDK o una CLI
  - Utilizzo DeleteUser con un AWS SDK o una CLI
  - Utilizzo ForgotPassword con un AWS SDK o una CLI
  - Utilizzo InitiateAuth con un AWS SDK o una CLI
  - Utilizzo ListUserPools con un AWS SDK o una CLI
  - Utilizzo ListUsers con un AWS SDK o una CLI
  - Utilizzo ResendConfirmationCode con un AWS SDK o una CLI
  - Utilizzo RespondToAuthChallenge con un AWS SDK o una CLI
  - Utilizzo SignUp con un AWS SDK o una CLI
  - Utilizzo UpdateUserPool con un AWS SDK o una CLI
  - Utilizzo VerifySoftwareToken con un AWS SDK o una CLI
- Scenari per Amazon Cognito Identity Provider che utilizzano SDK AWS
  - Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un SDK AWS
  - Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un SDK AWS
  - Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un SDK AWS

- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK AWS

Azioni per Amazon Cognito Identity Provider tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Identity Provider con AWS gli SDK. Questi estratti chiamano l'API Amazon Cognito Identity Provider e sono estratti di codice da programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per l'elenco completo, consulta Amazon Cognito Identity Provider API Reference (Documentazione di riferimento delle API del provider di identità di Amazon Cognito).

Esempi

- Utilizzo AdminCreateUser con un AWS SDK o una CLI
- Utilizzo AdminGetUser con un AWS SDK o una CLI
- Utilizzo AdminInitiateAuth con un AWS SDK o una CLI
- Utilizzo AdminRespondToAuthChallenge con un AWS SDK o una CLI
- Utilizzo AdminSetUserPassword con un AWS SDK o una CLI
- Utilizzo AssociateSoftwareToken con un AWS SDK o una CLI
- Utilizzo ConfirmDevice con un AWS SDK o una CLI
- Utilizzo ConfirmForgotPassword con un AWS SDK o una CLI
- Utilizzo ConfirmSignUp con un AWS SDK o una CLI
- Utilizzo CreateUserPool con un AWS SDK o una CLI
- Utilizzo CreateUserPoolClient con un AWS SDK o una CLI
- Utilizzo DeleteUser con un AWS SDK o una CLI
- Utilizzo ForgotPassword con un AWS SDK o una CLI
- Utilizzo InitiateAuth con un AWS SDK o una CLI
- Utilizzo ListUserPools con un AWS SDK o una CLI
- Utilizzo ListUsers con un AWS SDK o una CLI
- Utilizzo ResendConfirmationCode con un AWS SDK o una CLI

- Utilizzo RespondToAuthChallenge con un AWS SDK o una CLI
- Utilizzo SignUp con un AWS SDK o una CLI
- Utilizzo UpdateUserPool con un AWS SDK o una CLI
- Utilizzo VerifySoftwareToken con un AWS SDK o una CLI

Utilizzo **`AdminCreateUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AdminCreateUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito

CLI

AWS CLI

Per creare un utente

L'`admin-create-user`esempio seguente crea un utente con le impostazioni specificate: indirizzo e-mail e numero di telefono.

```
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
```

Output:

```
{
    "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"
    }
}
```

- Per i dettagli sull'API, vedere AdminCreateUserin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la AdminCreateUsersezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AdminGetUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AdminGetUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
   }
```

- Per i dettagli sull'API, consulta la AdminGetUsersezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AdminGetUsersezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per ottenere un utente

Questo esempio ottiene informazioni sul nome utente jane@example.com.

Comando:

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

Output:

```
{
  "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
```

```
}
```

- Per i dettagli sull'API, consulta AdminGetUser AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
       }
   }
```

- Per i dettagli sull'API, consulta la AdminGetUsersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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

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

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

- Per i dettagli sull'API, consulta la AdminGetUsersezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
    }
}
```

- Per i dettagli sull'API, AdminGetUserconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta AdminGetUser AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AdminInitiateAuth`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AdminInitiateAuth`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AdminInitiateAuthsezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AdminInitiateAuthsezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per avviare l'autorizzazione

Questo esempio avvia l'autorizzazione utilizzando il flusso ADMIN_NO_SRP_AUTH per il nome utente jane@example.com

Il client deve disporre dell'API di accesso per l'autenticazione basata sul server (ADMIN_NO_SRP_AUTH) abilitata.

Usa le informazioni sulla sessione nel valore restituito per chiamare admin-respond-to-auth - challenge.

Comando:

```
aws cognito-idp admin-initiate-auth --user-pool-id us-west-2_aaaaaaaaa --client-
id 3n4b5urk1ft4fl3mg5e62d9ado --auth-flow ADMIN_NO_SRP_AUTH --auth-parameters
 USERNAME=jane@example.com,PASSWORD=password
```

Output:

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

- Per i dettagli sull'API, consulta AdminInitiateAuth AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AdminInitiateAuthsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la AdminInitiateAuthsezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
    }
}
```

- Per i dettagli sull'API, AdminInitiateAuthconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.

        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
```

- Per i dettagli sull'API, consulta AdminInitiateAuth AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AdminRespondToAuthChallenge`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AdminRespondToAuthChallenge`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AdminRespondToAuthChallengesezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, consulta la AdminRespondToAuthChallengesezione AWS SDK for C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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());
    }
```

- Per i dettagli sull'API, consulta la AdminRespondToAuthChallengesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la AdminRespondToAuthChallengesezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
    }
}
```

- Per i dettagli sull'API, AdminRespondToAuthChallengeconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Rispondi a una richiesta di autenticazione MFA fornendo un codice generato da un'applicazione MFA associata.

```
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
```

- Per i dettagli sull'API, consulta AdminRespondToAuthChallenge AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AdminSetUserPassword`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`AdminSetUserPassword`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la AdminSetUserPasswordsezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`AssociateSoftwareToken`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`AssociateSoftwareToken`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la AssociateSoftwareTokensezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, consulta la AssociateSoftwareTokensezione AWS SDK for C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
   }
```

- Per i dettagli sull'API, consulta la AssociateSoftwareTokensezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const associateSoftwareToken = (session) => {
  const client = new CognitoIdentityProviderClient({});
  const command = new AssociateSoftwareTokenCommand({
    Session: session,
  });

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

- Per i dettagli sull'API, consulta la AssociateSoftwareTokensezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
    }
}
```

- Per i dettagli sull'API, AssociateSoftwareTokenconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta AssociateSoftwareToken AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ConfirmDevice`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmDevice`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la ConfirmDevicesezione AWS SDK for .NET API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la ConfirmDevicesezione AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ConfirmDevice AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ConfirmForgotPassword`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmForgotPassword`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda

CLI

AWS CLI

Per confermare una password dimenticata

Questo esempio conferma una password dimenticata per il nome utente diego@example.com.

Comando:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Per i dettagli sull'API, consulta ConfirmForgotPassword AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la ConfirmForgotPasswordsezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ConfirmSignUp`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ConfirmSignUp`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la ConfirmSignUpsezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, consulta la ConfirmSignUpsezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per confermare la registrazione

Questo esempio conferma la registrazione per il nome utente diego@example.com.

Comando:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- Per i dettagli sull'API, consulta ConfirmSignUp AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per i dettagli sull'API, consulta la ConfirmSignUpsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const confirmSignUp = ({ clientId, username, code }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ConfirmSignUpCommand({
    ClientId: clientId,
    Username: username,
    ConfirmationCode: code,
  });

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

- Per i dettagli sull'API, consulta la ConfirmSignUpsezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
    }
}
```

- Per i dettagli sull'API, ConfirmSignUpconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ConfirmSignUp AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateUserPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUserPool`.

CLI

AWS CLI

Creazione di un pool di utenti con configurazione minima

Questo esempio crea un pool di utenti denominato MyUserPool utilizzando valori predefiniti. Non ci sono attributi obbligatori né client di applicazioni. MFA e sicurezza avanzata sono disattivate.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Output:

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

Creazione di un pool di utenti con due attributi richiesti

Questo esempio crea un pool di utenti MyUserPool. Il pool è configurato per accettare l'e-mail come attributo del nome utente. Inoltre, imposta l'indirizzo e-mail di origine su un indirizzo convalidato utilizzando Amazon Simple Email Service (Amazon SES).

Comando:

```
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"
```

Output:

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

- Per i dettagli sull'API, vedere CreateUserPoolin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
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 "";
    }
}
```

- Per i dettagli sull'API, consulta la CreateUserPoolsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateUserPoolClient`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateUserPoolClient`.

CLI

AWS CLI

Per creare un client con pool di utenti

Questo esempio crea un nuovo client con pool di utenti con due flussi di autorizzazione espliciti: USER_PASSWORD_AUTH e ADMIN_NO_SRP_AUTH.

Comando:

```
aws cognito-idp create-user-pool-client --user-pool-id us-west-2_aaaaaaaaa
  --client-name MyNewClient --no-generate-secret --explicit-auth-flows
 "USER_PASSWORD_AUTH" "ADMIN_NO_SRP_AUTH"
```

Output:

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyNewClient",
      "ClientId": "6p3bs000no6a4ue1idruvd05ad",
      "LastModifiedDate": 1548697449.497,
      "CreationDate": 1548697449.497,
      "RefreshTokenValidity": 30,
      "ExplicitAuthFlows": [
          "USER_PASSWORD_AUTH",
          "ADMIN_NO_SRP_AUTH"
      ],
```

```
      "AllowedOAuthFlowsUserPoolClient": false
  }
}
```

- Per i CreateUserPoolClientdettagli AWS CLI sull'API, vedere in Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientReques
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CreateUserPoolClientRespon

/**
 * 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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la CreateUserPoolClientsezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteUser`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteUser`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Conferma automaticamente gli utenti conosciuti con una funzione Lambda
- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda
- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per eliminare un utente

Questo esempio elimina un utente.

Comando:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Per i dettagli sull'API, consulta DeleteUser AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la DeleteUsersezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ForgotPassword`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ForgotPassword`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda

CLI

AWS CLI

Per forzare la modifica della password

L'`forgot-password`esempio seguente invia un messaggio a jane@example.com per modificare la password.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --username
 jane@example.com
```

Output:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- Per i dettagli sull'API, vedere ForgotPasswordin AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la ForgotPasswordsezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`InitiateAuth`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`InitiateAuth`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Conferma automaticamente gli utenti conosciuti con una funzione Lambda
- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda
- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
```

```
    }
```

- Per i dettagli sull'API, consulta la InitiateAuthsezione AWS SDK for .NET API Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la InitiateAuthsezione AWS SDK for Go API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la InitiateAuthsezione AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo esempio mostra come avviare l'autenticazione con un dispositivo monitorato. Per completare l'accesso, il client deve rispondere correttamente alle richieste di autenticazione 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
```

- Per i dettagli sull'API, consulta InitiateAuth AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListUserPools`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUserPools`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per elencare i pool di utenti

Questo esempio elenca fino a 20 pool di utenti.

Comando:

```
aws cognito-idp list-user-pools --max-results 20
```

Output:

```
{
  "UserPools": [
      {
         "CreationDate": 1547763720.822,
         "LastModifiedDate": 1547763720.822,
         "LambdaConfig": {},
         "Id": "us-west-2_aaaaaaaaa",
         "Name": "MyUserPool"
      }
  ]
}
```

- Per i dettagli sull'API, consulta ListUserPools AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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%v: %v\n", *pool.Name, *pool.Id)
  }
 }
}
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListUserPoolssezione AWS SDK for Java 2.x API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la ListUserPoolsguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListUsers`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListUsers`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per elencare gli utenti

Questo esempio elenca fino a 20 utenti.

Comando:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Output:

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

```
}
```

- Per i dettagli sull'API, consulta ListUsers AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.cognitoidentityprovider.CognitoIdentityProviderClient;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.CognitoIdentityProviderExc
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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const listUsers = ({ userPoolId }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ListUsersCommand({
    UserPoolId: userPoolId,
  });

  return client.send(command);
```

```
};
```

- Per i dettagli sull'API, consulta la ListUserssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
        }
    }
}
```

- Per i dettagli sull'API, ListUsersconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ListUsers AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ResendConfirmationCode`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ResendConfirmationCode`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la ResendConfirmationCodesezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
       }
```

- Per i dettagli sull'API, consulta la ResendConfirmationCodesezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per inviare nuovamente un codice di conferma

L'esempio `resend-confirmation-code` seguente invia un codice di conferma all'utente `jane`.

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

Output:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

Per ulteriori informazioni, consulta Registrazione e conferma degli account utente nella Guida per gli sviluppatori di Amazon Cognito.

- Per i dettagli sull'API, consulta ResendConfirmationCode AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per i dettagli sull'API, consulta la ResendConfirmationCodesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const resendConfirmationCode = ({ clientId, username }) => {
  const client = new CognitoIdentityProviderClient({});

  const command = new ResendConfirmationCodeCommand({
    ClientId: clientId,
    Username: username,
  });

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

- Per i dettagli sull'API, consulta la ResendConfirmationCodesezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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))
    }
}
```

- Per i dettagli sull'API, ResendConfirmationCodeconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ResendConfirmationCode AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RespondToAuthChallenge`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RespondToAuthChallenge`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

CLI

AWS CLI

Per rispondere a una richiesta di autenticazione

Questo esempio risponde a una richiesta di autorizzazione avviata con initiate-auth. È una risposta alla richiesta NEW_PASSWORD_REQUIRED. Imposta una password per l'utente jane@example.com.

Comando:

```
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"
```

Output:

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

- Per i dettagli sull'API, consulta RespondToAuthChallenge AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la RespondToAuthChallengesezione AWS SDK for JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Accedi con un dispositivo monitorato. Per completare l'accesso, il client deve rispondere correttamente alle richieste di autenticazione 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
```

- Per i dettagli sull'API, consulta RespondToAuthChallenge AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`SignUp`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`SignUp`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Conferma automaticamente gli utenti conosciuti con una funzione Lambda
- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda
- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la SignUpsezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
```

- Per i dettagli sull'API, consulta la SignUpsezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per registrare un utente

In questo esempio viene registrato jane@example.com.

Comando:

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username jane@example.com --password PASSWORD --user-attributes
 Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Output:

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Per i dettagli sull'API, consulta SignUp AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la [SignUp](sezione AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per i dettagli sull'API, consulta la SignUpsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la SignUpsezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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")
    }
```

```
}
```

- Per i dettagli sull'API, SignUpconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta SignUp AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`UpdateUserPool`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`UpdateUserPool`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Conferma automaticamente gli utenti conosciuti con una funzione Lambda
- Esegui automaticamente la migrazione di utenti noti con una funzione Lambda
- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito

CLI

AWS CLI

Per aggiornare un pool di utenti

Questo esempio aggiunge tag a un pool di utenti.

Comando:

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_aaaaaaaaa --user-pool-
tags Team=Blue,Area=West
```

- Per i dettagli sull'API, consulta UpdateUserPool AWS CLICommand Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
}
```

- Per i dettagli sull'API, consulta la UpdateUserPoolsezione AWS SDK for Go API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`VerifySoftwareToken`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`VerifySoftwareToken`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Registrazione di un utente a un pool di utenti che richiede l'autenticazione MFA

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
```

- Per i dettagli sull'API, consulta la VerifySoftwareTokensezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
       }
```

- Per i dettagli sull'API, consulta la VerifySoftwareTokensezione AWS SDK for C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
```

- Per i dettagli sull'API, consulta la VerifySoftwareTokensezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
};
```

- Per i dettagli sull'API, consulta la VerifySoftwareTokensezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}")
    }
}
```

- Per i dettagli sull'API, VerifySoftwareTokenconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta VerifySoftwareToken AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per Amazon Cognito Identity Provider che utilizzano SDK AWS

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon Cognito Identity Provider con AWS SDK. Questi scenari illustrano come eseguire attività specifiche richiamando più funzioni del provider di identità di Amazon Cognito. Ogni scenario include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un SDK AWS
- Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un SDK AWS
- Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un SDK AWS
- Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK AWS

Conferma automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda utilizzando un SDK AWS

Il seguente esempio di codice mostra come confermare automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda.

- Configura un pool di utenti per chiamare una funzione Lambda per il `PreSignUp` trigger.
- Registra un utente con Amazon Cognito.
- La funzione Lambda analizza una tabella DynamoDB e conferma automaticamente gli utenti noti.
- Accedi come nuovo utente, quindi ripulisci le risorse.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
// 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))
}
```

Gestisci il `PreSignUp` grilletto con una funzione 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)
}
```

Crea una struttura che esegua attività comuni.

```
// 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))
}
```

Crea una struttura che racchiuda le azioni di 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
}
```

Crea una struttura che racchiuda le azioni di 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
}
```

Crea una struttura che racchiuda le azioni di Logs. CloudWatch

```
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
}
```

Crea una struttura che racchiuda le azioni. 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
}
```

Pulisci le risorse.

```
// 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!")
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - DeleteUser
  - InitiateAuth
  - SignUp
  - UpdateUserPool

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui automaticamente la migrazione di utenti Amazon Cognito noti con una funzione Lambda utilizzando un SDK AWS

Il seguente esempio di codice mostra come migrare automaticamente gli utenti noti di Amazon Cognito con una funzione Lambda.

- Configura un pool di utenti per chiamare una funzione Lambda per il `MigrateUser` trigger.
- Accedi ad Amazon Cognito con un nome utente e un indirizzo e-mail non inclusi nel pool di utenti.

- La funzione Lambda analizza una tabella DynamoDB e migra automaticamente gli utenti noti nel pool di utenti.
- Esegui il flusso relativo alla password dimenticata per reimpostare la password per l'utente migrato.
- Accedi come nuovo utente, quindi ripulisci le risorse.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
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))
}
```

Gestisci il `MigrateUser` grilletto con una funzione 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)
}
```

Crea una struttura che esegua attività comuni.

```
// 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))
}
```

Crea una struttura che racchiuda le azioni di 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
}
```

Crea una struttura che racchiuda le azioni di 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
}
```

Crea una struttura che racchiuda le azioni di Logs. CloudWatch

```
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
}
```

Crea una struttura che racchiuda le azioni. 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
}
```

Pulisci le risorse.

```
// 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!")
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - ConfirmForgotPassword
  - DeleteUser
  - ForgotPassword
  - InitiateAuth
  - SignUp

- UpdateUserPool

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Registra un utente con un pool di utenti Amazon Cognito che richiede l'autenticazione a più fattori utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Registra e conferma un utente con nome utente, password e indirizzo e-mail.
- Configura l'autenticazione a più fattori associando un'applicazione MFA all'utente.
- Accedi utilizzando una password e un codice MFA.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.Write("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.Write("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.Write("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;
    }

}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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();
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for C++ .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AdminRespondToAuthChalleng
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenRequ
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.AssociateSoftwareTokenResp
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.CognitoIdentityProviderExc
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ConfirmSignUpRequest;
import
 software.amazon.awssdk.services.cognitoidentityprovider.model.ResendConfirmationCodeRequ
```

```
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.VerifySoftwareTokenRespons
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("*** Conformation 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);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Per un'esperienza ottimale, clona il GitHub repository ed esegui questo esempio. Il codice seguente rappresenta un esempio dell'applicazione completa.

```
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);
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for JavaScript .
  - AdminGetUser

- AdminInitiateAuth
- AdminRespondToAuthChallenge
- AssociateSoftwareToken
- ConfirmDevice
- ConfirmSignUp
- InitiateAuth
- ListUsers
- ResendConfirmationCode
- RespondToAuthChallenge
- SignUp
- VerifySoftwareToken

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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 inOb = Scanner(System.`in`)
    val userName = inOb.nextLine()
    println(userName)

    println("*** Enter your password")
    val password: String = inOb.nextLine()

    println("*** Enter your email")
    val email = inOb.nextLine()
```

```
    println("*** Signing up $userName")
    signUp(clientId, userName, password, email)

    println("*** Getting $userName in the user pool")
    getAdminUser(userName, poolId)

    println("*** Conformation code sent to $userName. Would you like to send a
 new code? (Yes/No)")
    val ans = inOb.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 = inOb.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 = inOb.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 = inOb.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")
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Creazione di una classe che racchiude le funzioni di Amazon Cognito utilizzate nello scenario.

```
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
```

Creazione di una classe che esegue lo scenario. Questo esempio registra un dispositivo MFA che deve essere monitorato da Amazon Cognito e mostra come accedere utilizzando la password e le informazioni del dispositivo monitorato. evitando così di dover inserire un nuovo codice 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()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - AdminGetUser
  - AdminInitiateAuth
  - AdminRespondToAuthChallenge
  - AssociateSoftwareToken
  - ConfirmDevice
  - ConfirmSignUp
  - InitiateAuth
  - ListUsers
  - ResendConfirmationCode
  - RespondToAuthChallenge
  - SignUp
  - VerifySoftwareToken

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scrivi dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito tramite un SDK AWS

Il seguente esempio di codice mostra come scrivere dati di attività personalizzati con una funzione Lambda dopo l'autenticazione utente di Amazon Cognito.

- Usa le funzioni di amministratore per aggiungere un utente a un pool di utenti.
- Configura un pool di utenti per chiamare una funzione Lambda per il `PostAuthentication` trigger.
- Accedi il nuovo utente ad Amazon Cognito.

- La funzione Lambda scrive informazioni personalizzate nei CloudWatch log e in una tabella DynamoDB.
- Ottieni e visualizza dati personalizzati dalla tabella DynamoDB, quindi ripulisci le risorse.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
// 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.UserEmail)
 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))
}
```

Gestisci il `PostAuthentication` grilletto con una funzione 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)
}
```

Crea una struttura che esegua attività comuni.

```
// 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))
}
```

Crea una struttura che racchiuda le azioni di 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
```

```
}
```

Crea una struttura che racchiuda le azioni di 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
}
```

Crea una struttura che racchiuda le azioni di Logs. CloudWatch

```
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
}
```

Crea una struttura che racchiuda le azioni. 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
}
```

Pulisci le risorse.

```
// 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!")
  }
 }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - AdminCreateUser
  - AdminSetUserPassword

- DeleteUser
- InitiateAuth
- UpdateUserPool

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice per Amazon Cognito Sync tramite SDK AWS

I seguenti esempi di codice mostrano come usare Amazon Cognito Sync con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Azioni per Amazon Cognito Sync tramite SDK AWS
  - Utilizzo ListIdentityPoolUsage con un AWS SDK o una CLI

Azioni per Amazon Cognito Sync tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Cognito Sync con AWS gli SDK. Questi estratti chiamano l'API Amazon Cognito Sync e sono estratti di codice di programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per l'elenco completo, consulta Amazon Cognito Sync API Reference (Documentazione di riferimento delle API di Amazon Cognito Sync).

Esempi

- Utilizzo ListIdentityPoolUsage con un AWS SDK o una CLI

Utilizzo **`ListIdentityPoolUsage`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`ListIdentityPoolUsage`.

Rust

SDK per Rust

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la ListIdentityPoolUsageguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Best practice per le applicazioni multi-tenant

I pool di utenti di Amazon Cognito operano con applicazioni multi-tenant che generano un volume di richieste che deve rimanere entro le quote di Amazon Cognito. Per aumentare questa capacità man mano che la tua base di clienti cresce, puoi acquistare quote di capacità aggiuntive.

> **Note**
>
> Le quote di Amazon Cognito vengono applicate singolarmente. Account AWS Regione AWS Queste quote vengono condivise tra tutti i tenant dell'applicazione. Controlla le quote del servizio Amazon Cognito e assicurati che la quota soddisfi il volume previsto e il numero previsto di tenant nella tua applicazione.

Questa sezione descrive i metodi che puoi implementare per separare i tenant tra le risorse di Amazon Cognito all'interno della stessa regione e. Account AWS Puoi anche suddividere i tuoi inquilini in più di una Account AWS regione e assegnare a ciascuno di loro la propria quota. Altri vantaggi della multi-tenancy multiregionale includono il massimo livello di isolamento possibile, il tempo di transito di rete più breve per gli utenti distribuiti a livello globale e l'aderenza ai modelli di distribuzione esistenti nell'organizzazione.

La multi-tenancy a regione singola può avere vantaggi anche per i clienti e gli amministratori.

L'elenco seguente illustra alcuni dei vantaggi della multi-tenancy con risorse condivise.

Vantaggi della multi-tenancy

Elenco utenti comune

La multi-tenancy supporta modelli in cui i clienti dispongono di account in più di un'applicazione. È possibile collegare le identità di provider terzi in un unico profilo di pool di utenti coerente. Nei casi in cui i profili utente sono unici per il relativo tenant, qualsiasi strategia multi-tenancy con un unico pool di utenti prevede un unico punto di accesso per l'amministrazione degli utenti.

Sicurezza comune

In un pool di utenti condiviso, puoi creare un unico standard di sicurezza e applicare la stessa sicurezza avanzata, l'autenticazione a più fattori (MFA) AWS WAFe gli stessi standard a

tutti i tenant. Poiché un ACL AWS WAF Web deve coincidere con la risorsa a cui Regione AWS lo si associa, la multi-tenancy offre l'accesso condiviso a una risorsa complessa. Se desideri mantenere una configurazione di sicurezza coerente in applicazioni Amazon Cognito multiregionali, devi applicare standard operativi che replichino la configurazione tra le risorse.

Personalizzazione comune

Puoi personalizzare i pool di utenti e i pool di identità con AWS Lambda. La configurazione dei trigger Lambda nei pool di utenti e degli eventi di Amazon Cognito nei pool di identità può diventare complessa. Le funzioni Lambda devono trovarsi nello stesso pool Regione AWS di utenti o pool di identità. Le funzioni Lambda condivise possono applicare gli standard per i flussi di autenticazione personalizzati, la migrazione degli utenti, la generazione di token e altre funzioni all'interno di una regione.

Messaggistica comune

Amazon Simple Notification Service (Amazon SNS) richiede una configurazione aggiuntiva in una regione prima di poter inviare messaggi SMS ai tuoi utenti. Puoi inviare messaggi e-mail con identità e domini verificati di Amazon Simple Email Service (Amazon SES) contenuti in una regione.

Con la multi-tenancy, puoi condividere questo sovraccarico di configurazione e manutenzione tra tutti i tuoi tenant. Poiché Amazon SNS e Amazon SES non sono tutti disponibili Regioni AWS, la suddivisione delle risorse tra le regioni richiede ulteriori considerazioni.

Quando utilizzi provider di messaggistica personalizzati, ottieni la personalizzazione comune di una singola funzione Lambda per gestire la consegna dei messaggi.

L'interfaccia utente ospitata imposta un cookie di sessione nel browser in modo da riconoscere un utente che si è già autenticato. Quando autentichi gli utenti locali in un pool di utenti, il relativo cookie di sessione li autentica per tutti i client dell'app nello stesso pool di utenti. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno. Il cookie di sessione è valido per un'ora. Non è possibile modificare la durata del cookie di sessione.

Esistono due modi per impedire l'accesso tra client di app con un cookie di sessione dell'interfaccia utente ospitato.

- Separa i tuoi utenti in pool di utenti per tenant.
- Sostituisci l'accesso all'interfaccia utente ospitata con l'accesso all'API dei pool di utenti di Amazon Cognito.

Argomenti

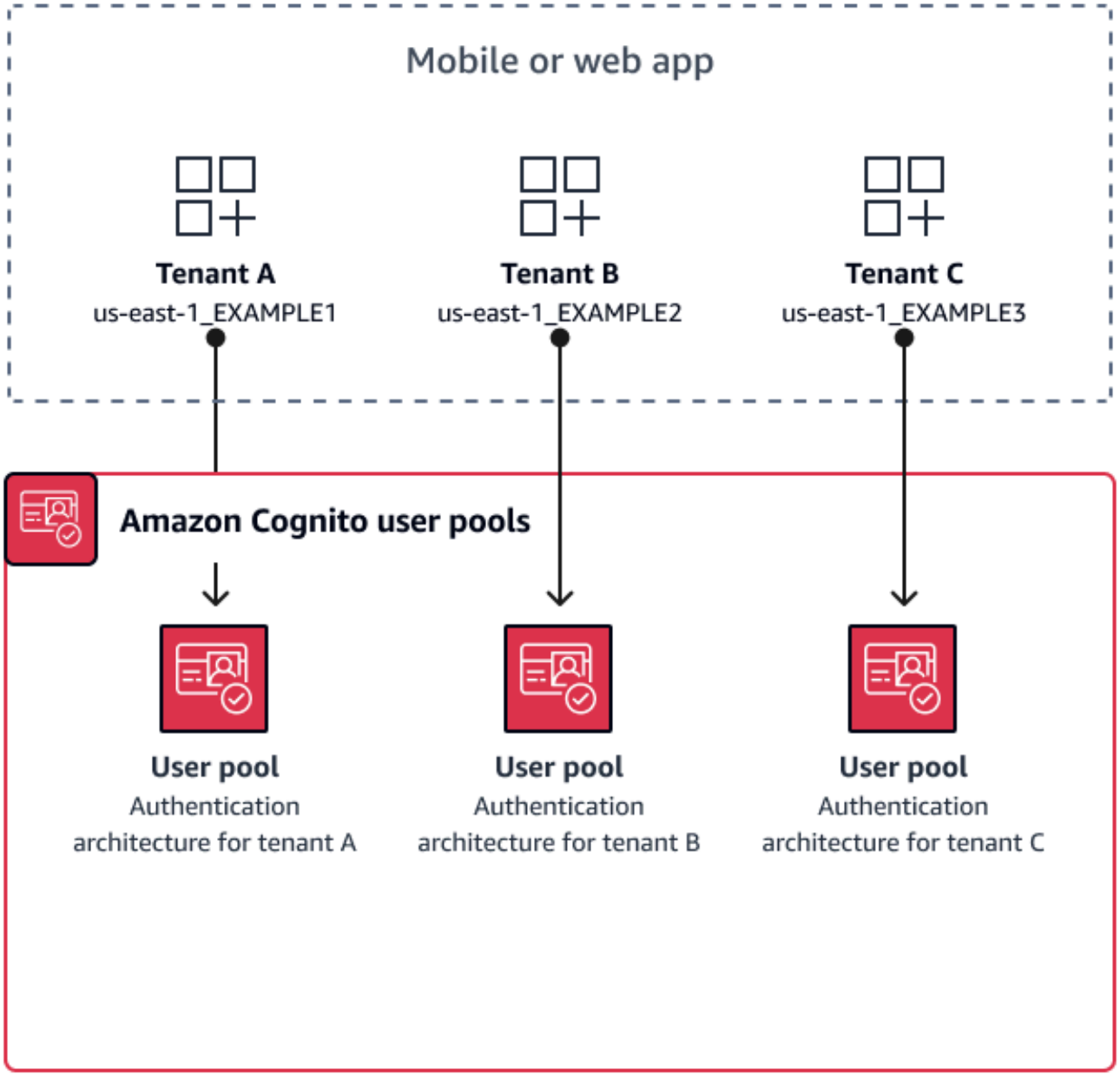
- Best practice multi-tenancy per pool di utenti
- Le migliori pratiche di multi-tenancy tra app e client
- Best practice multi-tenancy per gruppi di utenti
- Migliori pratiche di multi-tenancy con attributi personalizzati
- Suggerimenti per la sicurezza del multi-tenancy

Best practice multi-tenancy per pool di utenti

Crea un pool di utenti per ogni tenant nella tua app. Questo approccio fornisce il massimo isolamento per ogni tenant. È possibile implementare configurazioni diverse per ciascuno di essi. L'isolamento dei tenant per pool di utenti offre flessibilità nella user-to-tenant mappatura. È possibile creare più profili per lo stesso utente. Tuttavia, ogni utente deve registrarsi individualmente per ogni tenant a cui può accedere.

Utilizzando questo approccio, puoi configurare un'interfaccia utente ospitata per ogni tenant in modo indipendente e reindirizzare gli utenti all'istanza della tua applicazione specifica per il tenant. Puoi anche utilizzare questo approccio per l'integrazione con servizi di backend come Amazon API Gateway.

Il diagramma seguente mostra ogni tenant con un pool di utenti dedicato.



Quando implementare la modalità multi-tenancy per pool di utenti

Quando l'isolamento e la personalizzazione sono le tue preoccupazioni principali. La relazione tra utenti e tenant potrebbe essere complessa in un'architettura con più pool di utenti. Considerate un esempio in cui avete due inquilini che frequentano un istituto di istruzione. Lo stesso utente potrebbe essere uno studente con accesso limitato in un'app e un insegnante con un livello elevato di autorizzazioni in un'altra. Potresti richiedere la MFA in un'app ma non in un'altra o avere una politica

di password diversa. Poiché gli utenti locali possono accedere a più client di app nei pool di utenti con l'interfaccia utente ospitata, la modalità multi-tenancy con pool di utenti è ideale anche quando desideri che più di uno dei tuoi tenant acceda con l'interfaccia utente ospitata.

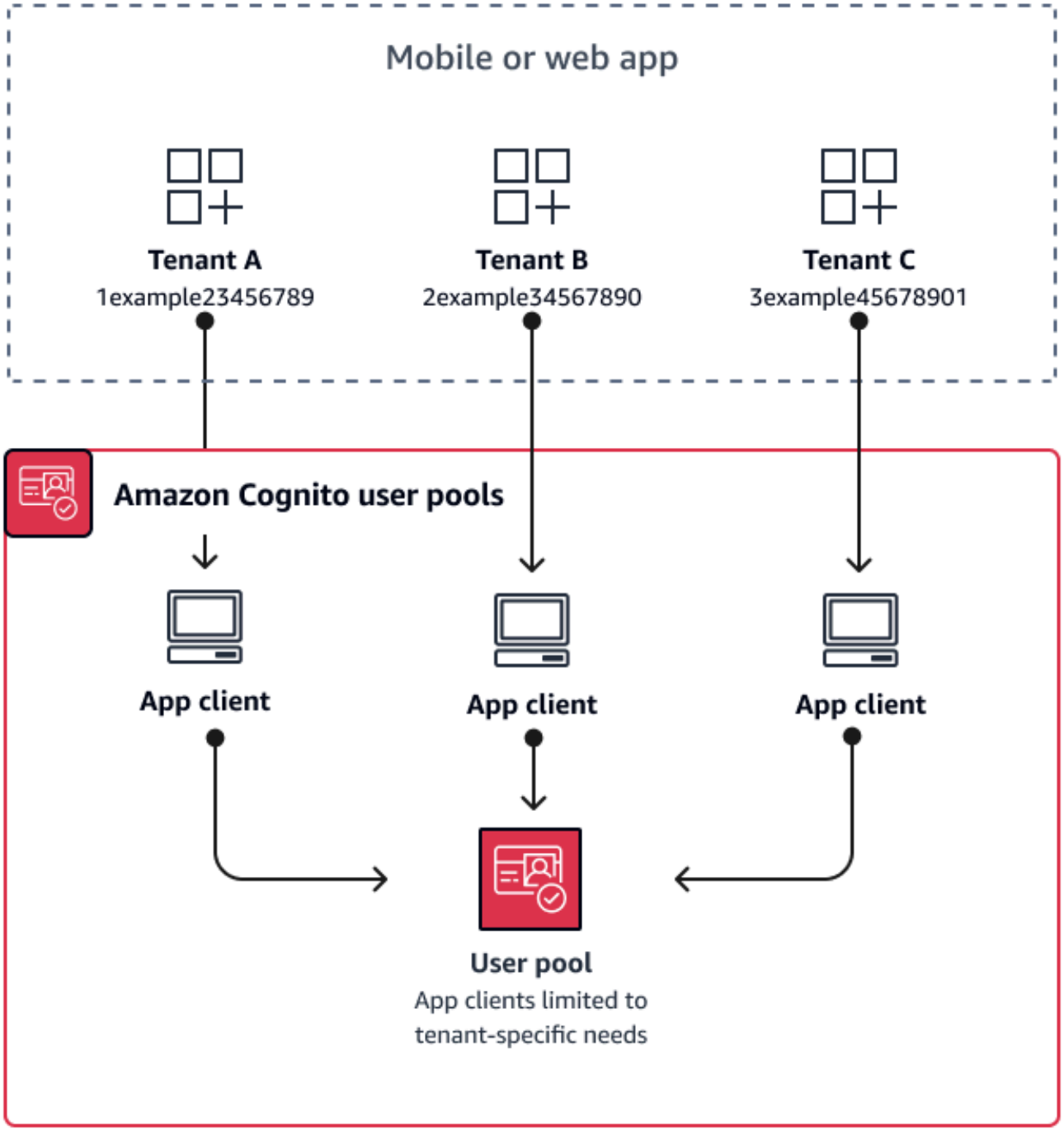
Livello di impegno

Lo sforzo di sviluppo e di funzionamento per utilizzare questo approccio è elevato. Per garantire risultati coerenti e prevedibili per la tua famiglia di app, devi integrare le risorse di Amazon Cognito con i tuoi strumenti di automazione e mantenere le tue linee di base man mano che l'architettura di autenticazione diventa più complessa. Quando vuoi creare un unico punto di partenza per le tue app, devi creare gli elementi dell'interfaccia utente (UI) per acquisire la decisione iniziale che indirizza gli utenti verso la risorsa corretta.

Le migliori pratiche di multi-tenancy tra app e client

Crea un app client per ogni tenant della tua app. Con app-client multi-tenancy, puoi assegnare qualsiasi utente a client di app collegati al tenant e mantenere un unico profilo utente. Poiché puoi assegnare uno o tutti gli identity provider (IdPs) del tuo pool di utenti a un app client, un client di app tenant può consentire l'accesso con un IdP specifico del tenant. Quando gli utenti esistono in più tenant, puoi collegare i loro profili a più tenant per un'esperienza utente coerente. IdPs

Il diagramma seguente mostra ogni tenant con un client di app dedicato in un pool di utenti condiviso.



Quando implementare la multi-tenancy tra app e client

Quando puoi scegliere una configurazione universale per le impostazioni a livello di pool di utenti, come i trigger Lambda, la politica delle password e il contenuto e i metodi di consegna di e-mail e

SMS. Poiché gli utenti di un pool di utenti condiviso possono accedere a qualsiasi client di app, la modalità multi-tenancy app-client è ideale per l'accesso con o con l'API dei pool di utenti di app-client-specific IdPs Amazon Cognito. La multi-tenancy app-client è ideale anche per one-to-many ambienti in cui si desidera consentire agli utenti di passare da un'applicazione all'altra.

Livello di impegno

La multi-tenancy tra app e client richiede uno sforzo moderato. Una delle principali sfide del multi-tenancy tra app e client è la possibilità per i tenant di presentare un cookie dell'interfaccia utente ospitato e passare da un'app all'altra. In un'architettura multi-tenancy tra app e client, evita l'accesso all'interfaccia utente ospitata dove è necessario l'isolamento. Puoi distribuire la tua app per dispositivi mobili o i link alla tua app web utilizzando la logica del client dell'app integrata, oppure puoi creare elementi iniziali dell'interfaccia utente che determinano la locazione degli utenti. Il livello di impegno è inferiore perché non è necessario standardizzare e mantenere la configurazione su più pool di utenti e pool di identità.

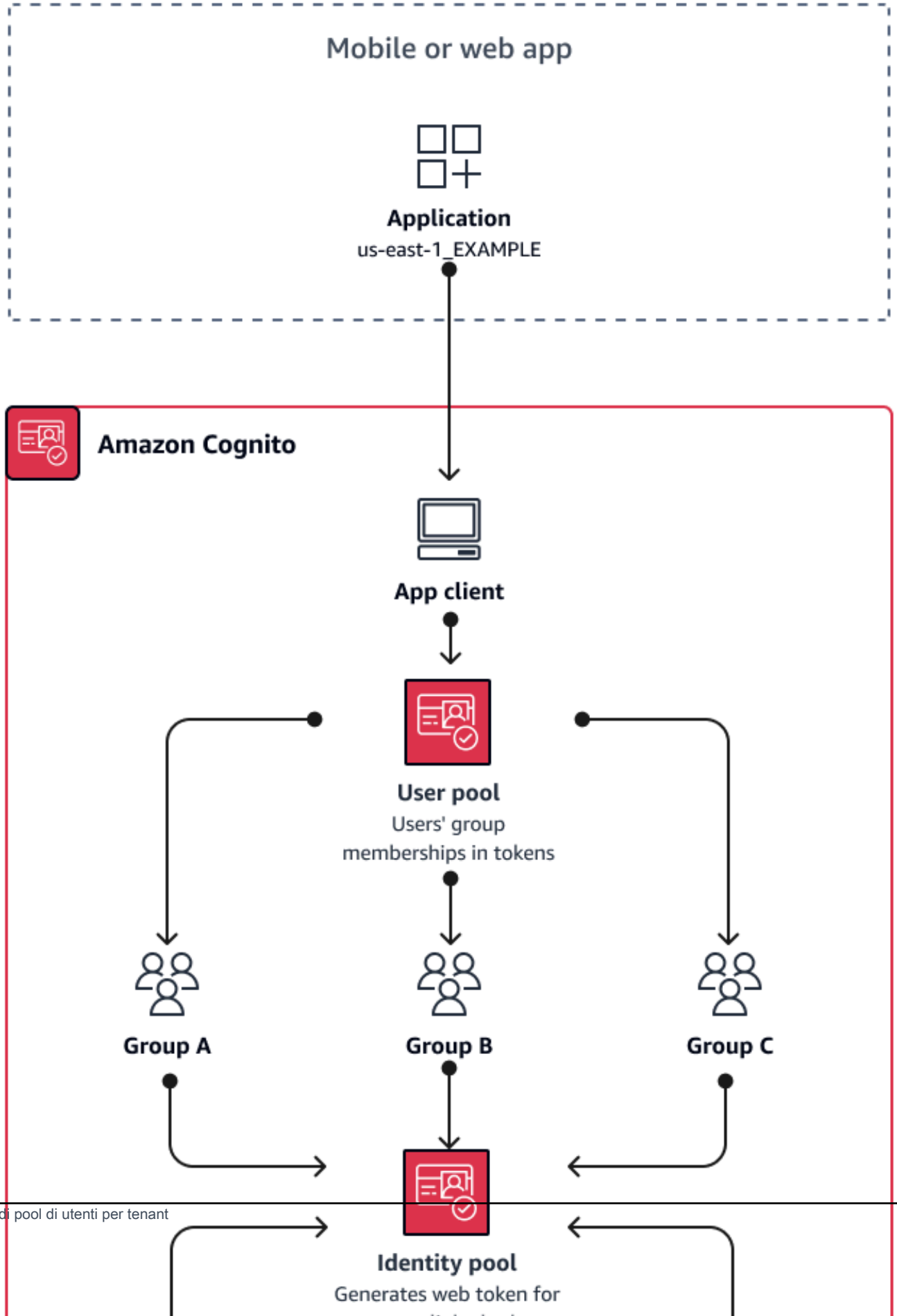
Best practice multi-tenancy per gruppi di utenti

La multi-tenancy basata su gruppi funziona meglio quando la tua architettura richiede pool di utenti Amazon Cognito con pool di identità.

L'ID del pool di utenti e i token di accesso contengono un claim. `cognito:groups` Inoltre, i token ID contengono `cognito:roles` e `cognito:preferred_role` rivendicazioni. Quando il risultato principale dell'autenticazione nella tua app sono AWS credenziali temporanee provenienti da un pool di identità, l'appartenenza ai gruppi degli utenti può determinare il ruolo IAM e le autorizzazioni che ricevono.

Ad esempio, prendiamo in considerazione tre tenant, ciascuno dei quali archivia gli asset applicativi nel proprio bucket Amazon S3. Assegna gli utenti di ogni tenant a un gruppo associato, configura un ruolo preferito per il gruppo e concedi a quel ruolo l'accesso in lettura al relativo bucket.

Il diagramma seguente mostra i tenant che condividono un client di app e un pool di utenti, con gruppi dedicati nel pool di utenti che determinano la loro idoneità per un ruolo IAM.


Mobile or web app
Application
us-east-1_EXAMPLE
Amazon Cognito
App client
User pool
Users' group
memberships in tokens
Group A
Group B
Group C
Identity pool
Generates web token for

Quando implementare la multi-tenancy di gruppo

Quando l'accesso alle AWS risorse è la tua preoccupazione principale. I gruppi nei pool di utenti di Amazon Cognito sono un meccanismo per il controllo degli accessi basato sui ruoli (RBAC). Puoi configurare molti gruppi in un pool di utenti e prendere decisioni RBAC complesse con priorità di gruppo. I pool di identità possono assegnare credenziali per il ruolo con la priorità più alta, per qualsiasi ruolo nel claim di gruppo o per altre attestazioni nei token di un utente.

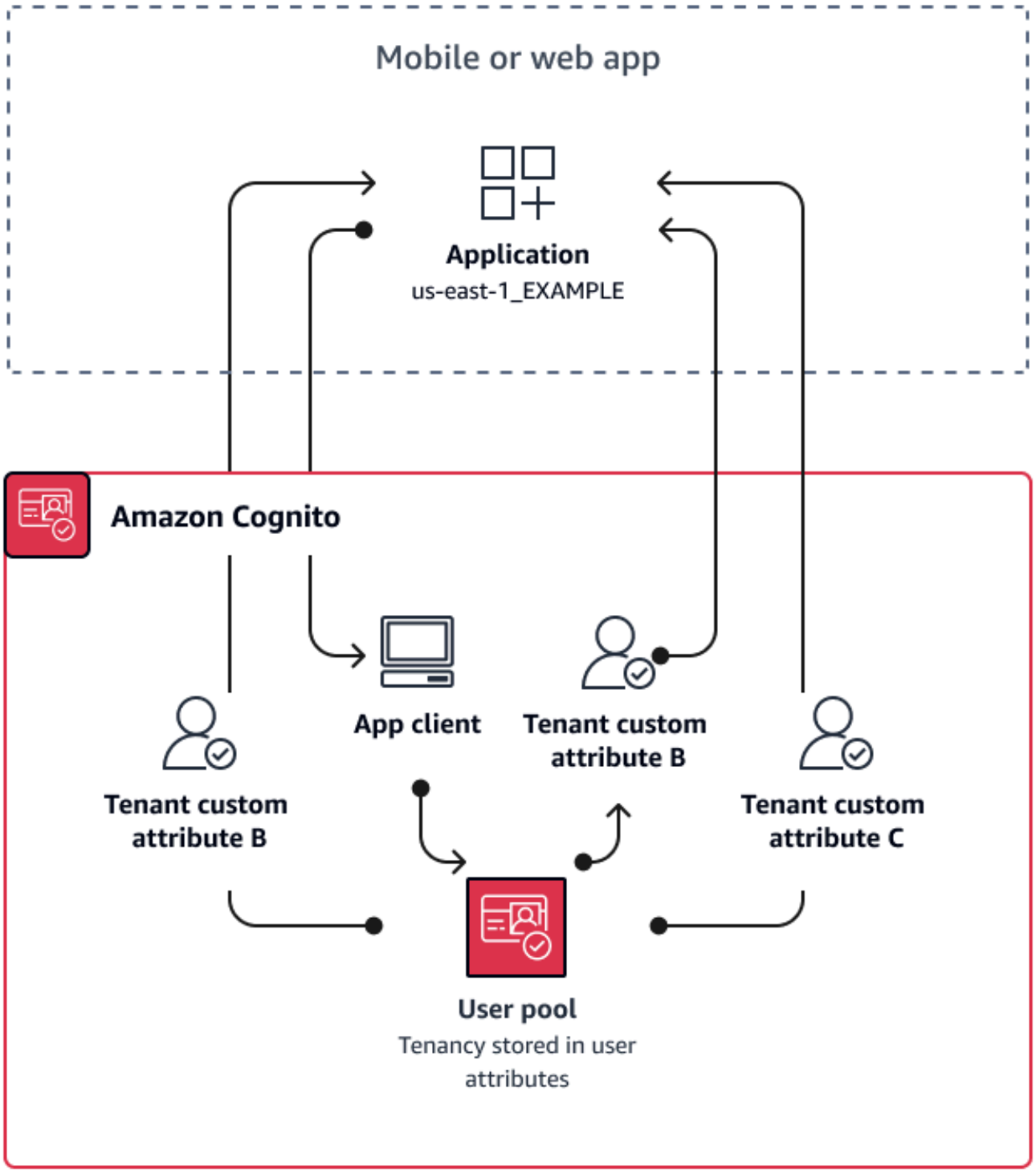
Livello di impegno

Il livello di impegno per mantenere la multi-tenancy con la sola appartenenza al gruppo è basso. Tuttavia, per ampliare il ruolo dei gruppi di pool di utenti oltre alla capacità integrata di selezione dei ruoli IAM, è necessario creare una logica applicativa che elabori l'appartenenza ai gruppi nei token degli utenti e determinare cosa fare nel client. Puoi integrare Amazon Verified Permissions con le tue app per prendere decisioni di autorizzazione lato client. Gli identificatori di gruppo non vengono attualmente elaborati nelle operazioni dell'IsAuthorizedWithTokenAPI Verified Permissions, ma puoi sviluppare codice personalizzato che analizzi il contenuto dei token, comprese le richieste di appartenenza ai gruppi.

Migliori pratiche di multi-tenancy con attributi personalizzati

Amazon Cognito supporta attributi personalizzati con nomi a tua scelta. Uno scenario in cui gli attributi personalizzati sono utili è quando distinguono la locazione degli utenti in un pool di utenti condiviso. Quando assegni agli utenti un valore per un attributo come`custom:tenantID`, ad esempio, la tua app può assegnare di conseguenza l'accesso alle risorse specifiche del tenant. Un attributo personalizzato che definisce un ID tenant deve essere immutabile o di sola lettura per il client dell'app.

Il diagramma seguente mostra i tenant che condividono un client di app e un pool di utenti, con attributi personalizzati nel pool di utenti che indica il tenant a cui appartengono.



Quando gli attributi personalizzati determinano la locazione, è possibile distribuire una singola applicazione o un URL di accesso. Dopo che l'utente ha effettuato l'accesso, l'app può elaborare

il `custom:tenantID` reclamo, determinare quali risorse caricare, il marchio da applicare e le funzionalità da visualizzare. Per decisioni avanzate sul controllo degli accessi in base agli attributi degli utenti, configura il tuo pool di utenti come provider di identità in Amazon Verified Permissions e genera decisioni di accesso dal contenuto di ID o token di accesso.

Quando implementare la multi-tenancy con attributi personalizzati

Quando la locazione è a livello di superficie. Un attributo tenant può contribuire ai risultati di branding e layout. Quando si desidera ottenere un isolamento significativo tra i tenant, gli attributi personalizzati non sono la scelta migliore. Qualsiasi differenza tra i tenant che devono essere configurati a livello di pool di utenti o app-client, come MFA o il branding dell'interfaccia utente ospitata, richiede la creazione di distinzioni tra i tenant in un modo che gli attributi personalizzati non offrono. Con i pool di identità, puoi persino scegliere il ruolo IAM tra i tuoi utenti dall'indicazione degli attributi personalizzati contenuta nel loro token ID.

Livello di impegno

Poiché la multi-tenancy con attributi personalizzati trasferisce l'obbligo di prendere decisioni di autorizzazione basate sul tenant sulla vostra app, il livello di impegno tende ad essere elevato. Se sei già esperto in una configurazione client che analizza le dichiarazioni OIDC o in Amazon Verified Permissions, questo approccio potrebbe richiedere il minimo sforzo.

Suggerimenti per la sicurezza del multi-tenancy

Per rendere la tua applicazione più sicura, ti consigliamo di seguire questi suggerimenti:

- Convalida la locazione nella tua app con Amazon Verified Permissions. Crea politiche che esaminino l'autorizzazione del pool di utenti, del client dell'app, del gruppo o degli attributi personalizzati prima di consentire la richiesta di un utente nella tua applicazione. AWS ha creato fonti di identità Verified Permissions pensando ai pool di utenti di Amazon Cognito. Verified Permissions offre ulteriori linee guida per la gestione multi-tenancy.
- Usa solo un indirizzo e-mail verificato per autorizzare l'accesso degli utenti a un tenant in base alla corrispondenza del dominio. Non fidarti degli indirizzi e-mail e dei numeri di telefono a meno che la tua app non li verifichi o che l'IdP esterno fornisca una prova di verifica. Per maggiori dettagli sull'impostazione di queste autorizzazioni, consulta la sezione Autorizzazioni e ambiti degli attributi.
- Utilizza attributi personalizzati immutabili o di sola lettura per gli attributi del profilo utente che identificano i tenant. Puoi impostare il valore degli attributi immutabili solo quando crei un utente

o quando un utente si iscrive al tuo pool di utenti. Fornisci inoltre ai client di app l'accesso in sola lettura agli attributi.

- Utilizza la mappatura 1:1 tra l'IdP esterno di un tenant e il client dell'applicazione per impedire l'accesso non autorizzato tra tenant. Un utente autenticato da un IdP esterno e dotato di un cookie di sessione Amazon Cognito valido, può accedere ad altre App tenant che considerano lo stesso IdP come affidabile.
- Quando implementi la logica di autorizzazione e corrispondenza tenant nell'applicazione, imposta delle limitazioni affinché le autorizzazioni di accesso degli utenti ai tenant non possano essere modificate dagli utenti stessi. Inoltre, se per la federazione viene utilizzato un IdP esterno, limita gli amministratori del provider di identità del tenant in modo che non possano modificare l'accesso degli utenti.

Scenari comuni di Amazon Cognito

Questo argomento descrive sei scenari comuni per l'utilizzo di Amazon Cognito.

I due componenti principali di Amazon Cognito sono i bacini d'utenza e i pool di identità. I bacini d'utenza sono directory utente che forniscono opzioni di registrazione e di accesso agli utenti delle tue app Web e per dispositivi mobili. I pool di identità forniscono AWS credenziali temporanee per concedere agli utenti l'accesso ad altri Servizi AWS.

Un bacino d'utenza è una directory di utenti in Amazon Cognito. Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Sia se gli utenti effettuano l'accesso direttamente o tramite terze parti, tutti i membri del bacino d'utenza dispongono di un profilo di directory a cui è possibile accedere tramite un SDK.

Con un pool di identità, i tuoi utenti possono ottenere AWS credenziali temporanee per accedere a AWS servizi come Amazon S3 e DynamoDB. I pool di identità supportano utenti ospiti anonimi e la federazione tramite terze parti. IdPs

Argomenti

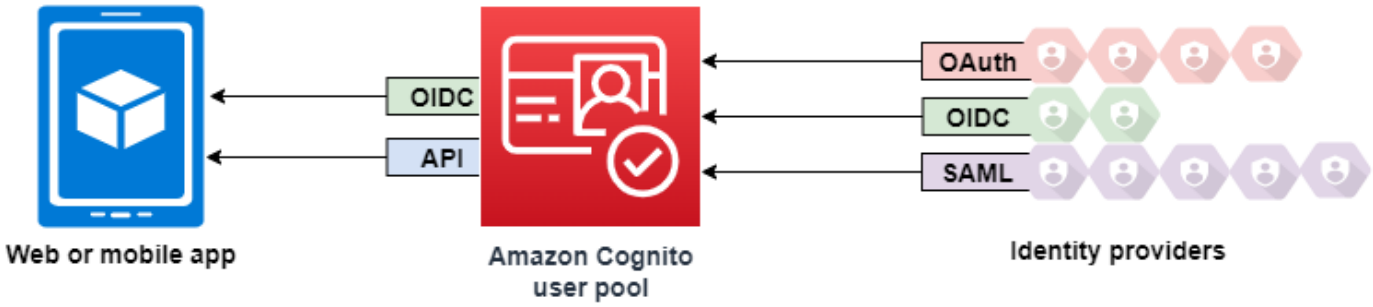
- Autenticazione con un bacino d'utenza
- Accesso alle risorse lato server con un bacino d'utenza
- Accesso alle risorse con API Gateway e Lambda tramite un bacino d'utenza
- Accedi ai AWS servizi con un pool di utenti e un pool di identità
- Autenticazione con terze parti e accesso ai servizi AWS con un pool di identità
- Accedi alle AWS AppSync risorse con Amazon Cognito

Autenticazione con un bacino d'utenza

Puoi consentire ai tuoi utenti di autenticarsi con un bacino d'utenza. Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs

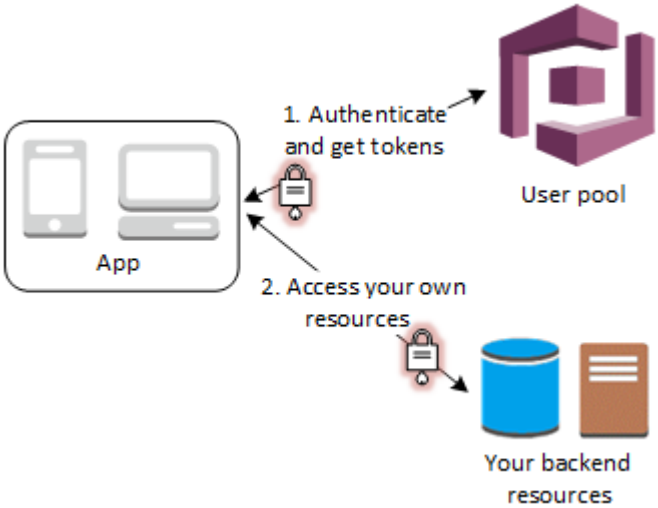
Dopo una corretta autenticazione, l'app Web o per dispositivi mobili riceverà i token del bacino d'utenza da Amazon Cognito. Puoi utilizzare questi token per recuperare AWS le credenziali che consentono alla tua app di accedere ad altri AWS servizi oppure puoi scegliere di usarli per controllare l'accesso alle tue risorse lato server o ad Amazon API Gateway.

Per ulteriori informazioni, consulta Flusso di autenticazione del bacino d'utenza e Utilizzo di token con bacini d'utenza.



Accesso alle risorse lato server con un bacino d'utenza

Dopo aver eseguito l'accesso al bacino d'utenza, l'applicazione Web o per dispositivi mobili riceverà i token del bacino d'utenza da Amazon Cognito. È possibile utilizzare questi token per controllare l'accesso alle risorse lato server. Puoi anche creare gruppi di bacini d'utenza per gestire le autorizzazioni o rappresentare diversi tipi di utenti. Per ulteriori informazioni sull'utilizzo dei gruppi per controllare l'accesso alle risorse, consulta Aggiunta di gruppi a un bacino d'utenza.



Dopo aver configurato un dominio per il pool di utenti, Amazon Cognito effettua il provisioning di un'interfaccia utente Web ospitata che consente di aggiungere pagine di registrazione e di accesso

all'app. Tramite questa base OAuth 2.0 è possibile creare il server di risorse per consentire agli utenti di accedere alle risorse protette. Per ulteriori informazioni, consulta Autorizzazione Scopes, M2M e API con server di risorse.

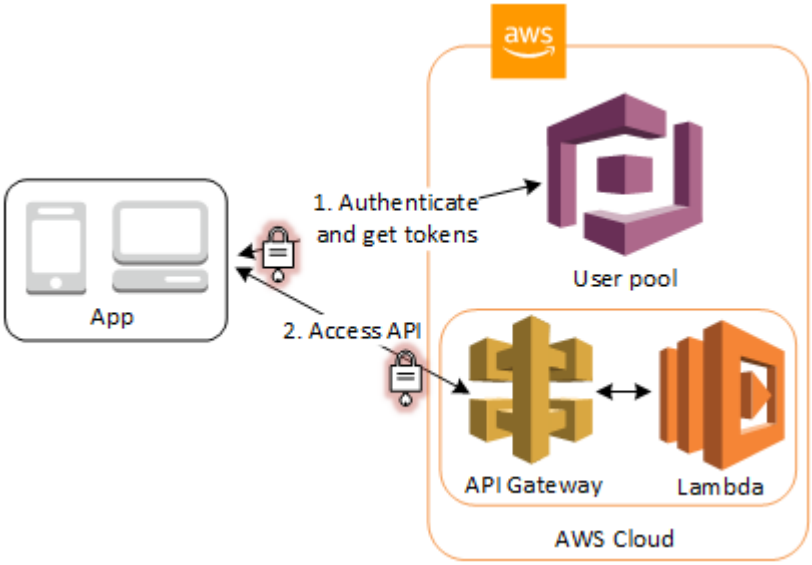
Per ulteriori informazioni sull'autenticazione dei pool di utenti, consulta Flusso di autenticazione del bacino d'utenza e Utilizzo di token con bacini d'utenza.

Accesso alle risorse con API Gateway e Lambda tramite un bacino d'utenza

È possibile consentire agli utenti di accedere all'API tramite API Gateway. API Gateway convalida i token da una corretta autenticazione del bacino d'utenza e li utilizza per concedere agli utenti l'accesso a risorse, tra cui le funzioni Lambda o la tua API.

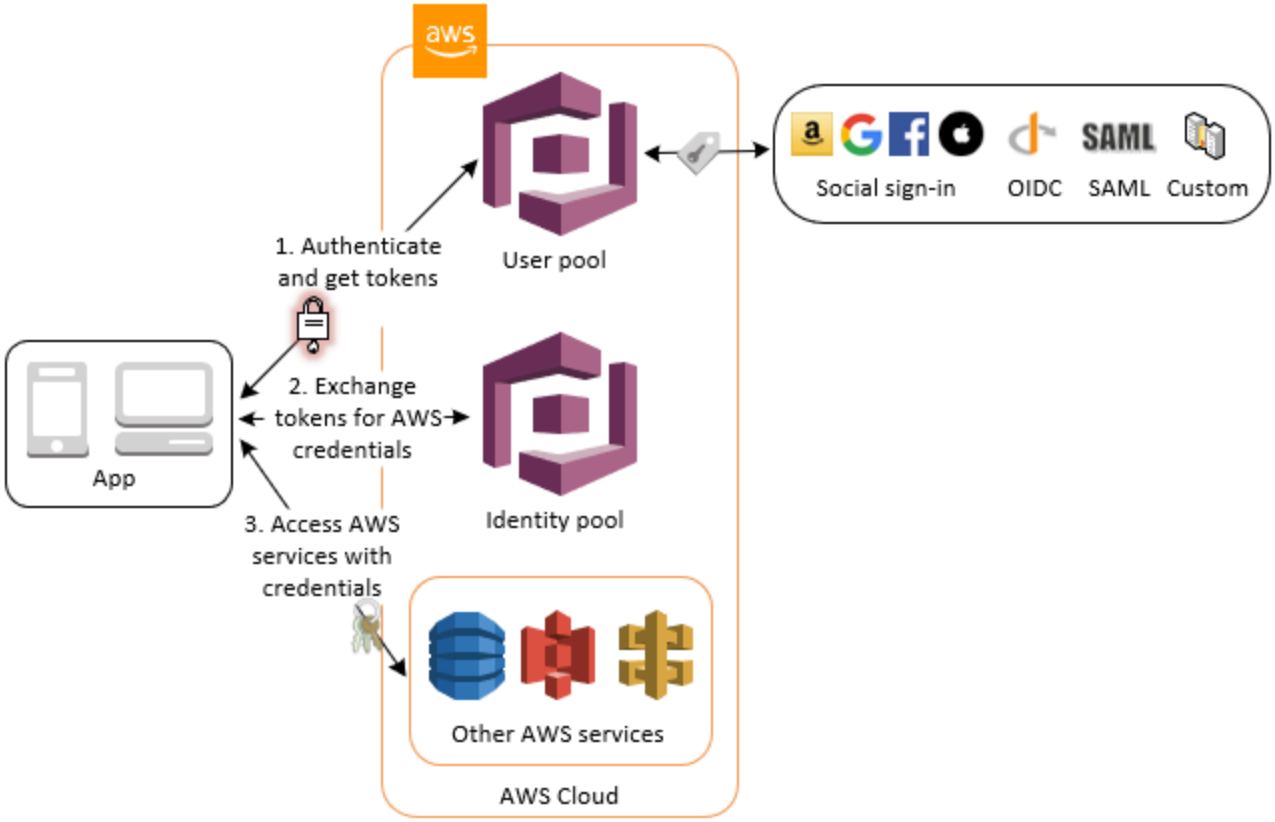
È possibile utilizzare i gruppi in un bacino d'utenza per controllare le autorizzazioni con API Gateway mappando l'appartenenza al gruppo ai ruoli IAM. I gruppi di cui un utente è membro sono inclusi nel token ID fornito da un bacino d'utenza quando l'utente dell'app effettua l'accesso. Per ulteriori informazioni sui gruppi di bacini d'utenza, consulta Aggiunta di gruppi a un bacino d'utenza.

È possibile inviare i token del bacino d'utenza con una richiesta ad API Gateway di verifica mediante una funzione Lambda di autorizzazione di Amazon Cognito. Per ulteriori informazioni su API Gateway, consulta Utilizzo di API Gateway con bacini d'utenza di Amazon Cognito.



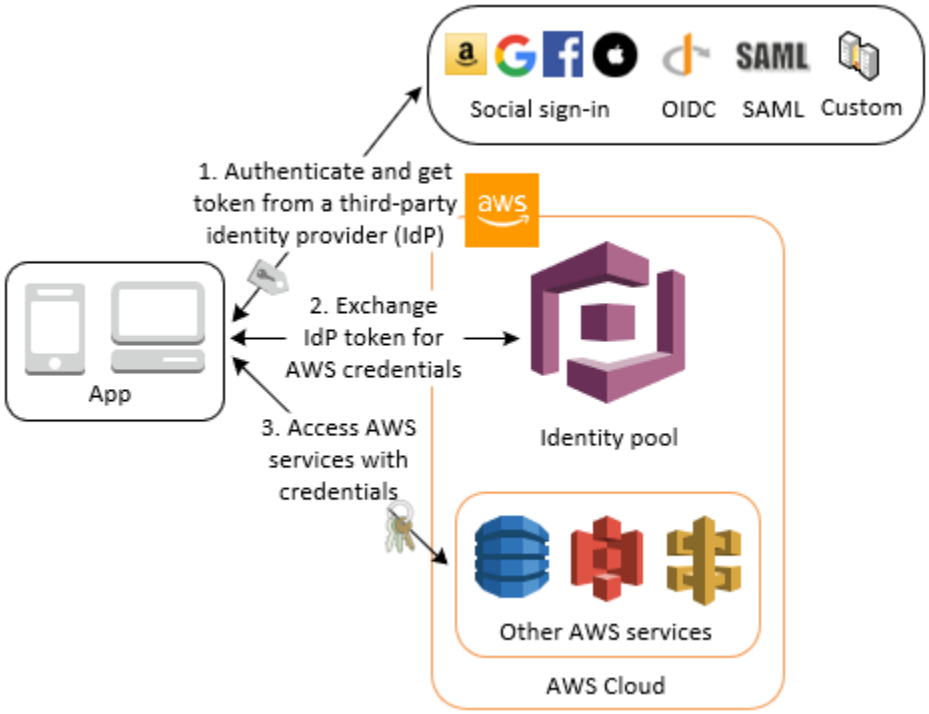
Accedi ai AWS servizi con un pool di utenti e un pool di identità

Dopo una corretta autenticazione del bacino d'utenza, l'app riceverà i token del bacino d'utenza da Amazon Cognito. Puoi scambiarli con l'accesso temporaneo ad altri AWS servizi con un pool di identità. Per ulteriori informazioni, consulta Accesso Servizi AWS tramite un pool di identità dopo l'accesso e Guida introduttiva ai pool di identità di Amazon Cognito.



Autenticazione con terze parti e accesso ai servizi AWS con un pool di identità

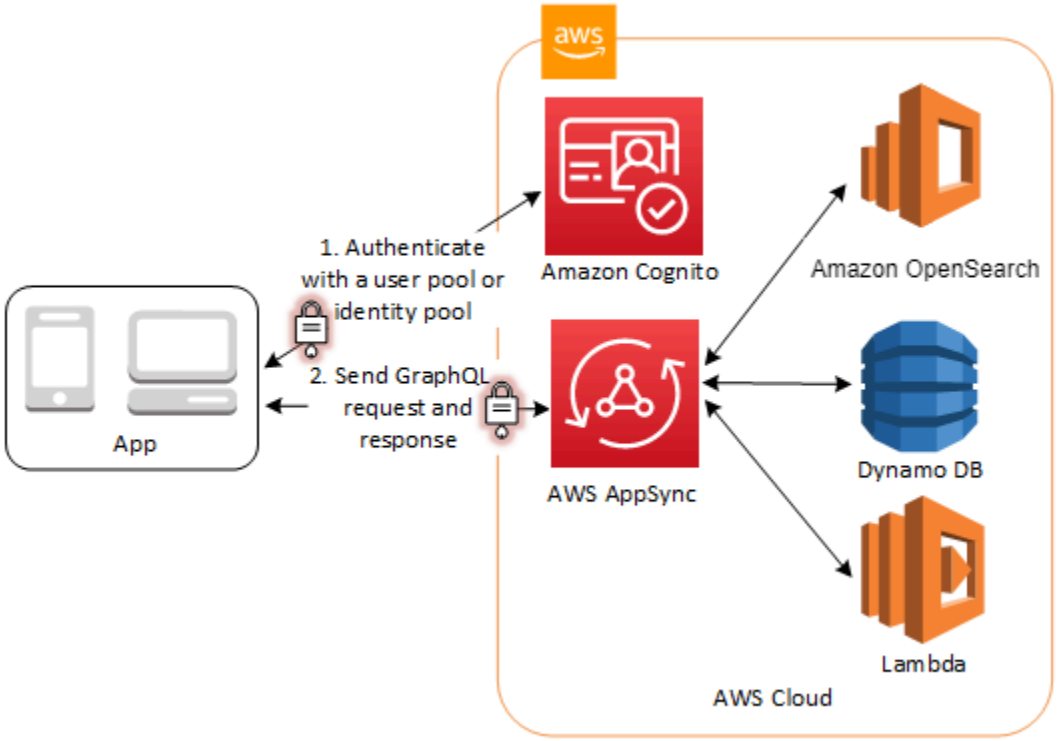
È possibile consentire agli utenti di accedere ai AWS servizi tramite un pool di identità. Un pool di identità richiede un token IdP da un utente autenticato da un provider di identità di terze parti (o niente se è un utente guest anonimo). In cambio, il pool di identità concede AWS credenziali temporanee che è possibile utilizzare per accedere ad altri AWS servizi. Per ulteriori informazioni, consulta Guida introduttiva ai pool di identità di Amazon Cognito.



Accedi alle AWS AppSync risorse con Amazon Cognito

Puoi concedere ai tuoi utenti l'accesso alle AWS AppSync risorse con i token di un'autenticazione riuscita del pool di utenti di Amazon Cognito. Per ulteriori informazioni, consultare Autorizzazione AMAZON_COGNITO_USER_POOLS nella Guida per sviluppatori di AWS AppSync .

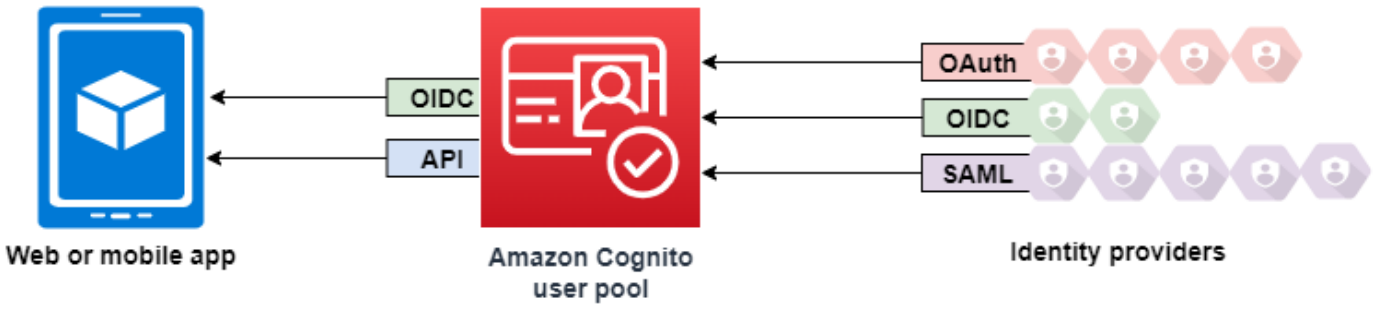
Puoi anche firmare le richieste all'API AWS AppSync GraphQL con le credenziali IAM che ricevi da un pool di identità. Consultare Autorizzazione AWS_IAM.


aws
Amazon Cognito
Amazon OpenSearch
1. Authenticate with a user pool or identity pool
2. Send GraphQL request and response
App
AWS AppSync
Dynamo DB
Lambda
AWS Cloud

Bacini d'utenza di Amazon Cognito

Un pool di utenti Amazon Cognito è una directory utente per l'autenticazione e l'autorizzazione di app web e per dispositivi mobili. Dal punto di vista dell'app, un pool di utenti Amazon Cognito è un gestore dell'identità digitale OpenID Connect (OIDC). Un pool di utenti aggiunge ulteriori livelli di funzionalità per la sicurezza, la federazione delle identità, l'integrazione app e la personalizzazione dell'esperienza utente.

Puoi, ad esempio, verificare che le sessioni degli utenti provengano da fonti attendibili. Puoi combinare la directory Amazon Cognito con un provider di identità esterno. Con il tuo AWS SDK preferito, puoi scegliere il modello di autorizzazione API più adatto alla tua app. Inoltre, puoi aggiungere funzioni AWS Lambda che modificano o riorganizzano il comportamento predefinito di Amazon Cognito.



Argomenti

- Funzionalità
- Autenticazione con un bacino d'utenza
- Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti
- Aggiornamento della configurazione del pool di utenti
- Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione
- Autorizzazione Scopes, M2M e API con server di risorse
- Aggiunta di un accesso al bacino d'utenza tramite terze parti
- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.
- Gestione degli utenti nel tuo bacino d'utenza

- Impostazioni e-mail per i bacini d'utenza di Amazon Cognito
- Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito
- Utilizzo di token con bacini d'utenza
- Accesso alle risorse dopo una corretta autenticazione del bacino d'utenza
- Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito

Funzionalità

Di seguito sono riportate le funzionalità dei pool di utenti Amazon Cognito.

Registrazione

I pool di utenti Amazon Cognito dispongono di metodi programmatici, basati sugli utenti e sugli amministratori per aggiungere profili utente al pool di utenti. I pool di utenti Amazon Cognito supportano i seguenti modelli di registrazione. Puoi usare qualsiasi combinazione di questi modelli nell'app.

> **Important**
>
> Se attivi la registrazione dell'utente nel pool di utenti, chiunque su Internet può effettuare la registrazione a un account e accedere alle tue app. Non abilitare la registrazione self-service nel pool di utenti a meno che non desideri aprire l'app alla registrazione pubblica. Per modificare questa impostazione, aggiorna l'iscrizione in modalità self-service nella scheda Esperienza di registrazione della console del pool di utenti o aggiorna il valore di AllowAdminCreateUserOnlyin una CreateUserPoolrichiesta o API. UpdateUserPool Per informazioni sulle funzionalità di sicurezza che puoi configurare nei pool di utenti, consulta Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito.

1. Gli utenti possono inserire le proprie informazioni nell'app e creare un profilo utente nativo per il pool di utenti. Puoi chiamare operazioni di registrazione delle API per registrare gli utenti nel pool di utenti. Puoi aprire queste operazioni di registrazione a chiunque oppure puoi autorizzarle con un segreto o credenziali del client. AWS
2. Puoi reindirizzare gli utenti a un IdP di terze parti che può essere autorizzato a passare le informazioni ad Amazon Cognito. Amazon Cognito elabora i token ID OIDC, i

dati `userInfo` OAuth 2.0 e le asserzioni SAML 2.0 nei profili utente del pool di utenti. Puoi controllare gli attributi che deve ricevere Amazon Cognito in base alle regole di mappatura degli attributi.

3. Puoi saltare la registrazione pubblica o federata e creare utenti in base all'origine dati e allo schema. Aggiungi gli utenti direttamente nella console di Amazon Cognito o nell'API. Importa utenti da un file CSV. Esegui una just-in-time AWS Lambda funzione che cerchi il nuovo utente in una directory esistente e compili il suo profilo utente con i dati esistenti.

Dopo la registrazione, è possibile aggiungerli ai gruppi elencati da Amazon Cognito nei token di accesso e ID. Puoi anche collegare i gruppi di pool di utenti ai ruoli IAM quando passi il token ID a un pool di identità.

Argomenti correlati

- Gestione degli utenti nel tuo bacino d'utenza
- Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti
- Esempi di codice per Amazon Cognito Identity Provider che utilizza SDK AWS

Accesso

Amazon Cognito può essere una directory utente autonoma e gestore dell'identità digitale per l'app. Gli utenti possono accedere con un'interfaccia utente ospitata da Amazon Cognito o con la propria interfaccia utente tramite l'API dei pool di utenti Amazon Cognito. Il livello dell'applicazione dietro l'interfaccia utente personalizzata del front-end può autorizzare le richieste sul back-end con uno di diversi metodi per confermare le richieste legittime.

Per accedere agli utenti con una directory esterna, facoltativamente combinata con la directory utente integrata in Amazon Cognito, puoi aggiungere le seguenti integrazioni.

1. Accedi e importa i dati degli utenti con l'accesso social OAuth 2.0. Amazon Cognito supporta l'accesso con Google, Facebook, Amazon e Apple tramite OAuth 2.0.
2. Accedi e importa i dati degli utenti aziendali con l'accesso SAML e OIDC. Puoi anche configurare Amazon Cognito per accettare richieste da qualsiasi gestore dell'identità digitale SAML o OpenID Connect (OIDC).
3. Collega i profili utente esterni ai profili utente nativi. Un utente collegato può accedere con un'identità utente di terze parti e ricevere l'accesso assegnato a un utente nella directory integrata.

Argomenti correlati

- Aggiunta di un accesso al bacino d'utenza tramite terze parti
- Collegamento di utenti federati a un profilo utente esistente

Autorizzazione M achine-to-machine

Alcune sessioni non sono un' human-to-machine interazione. Potrebbe essere necessario un account di servizio in grado di autorizzare una richiesta a un'API tramite un processo automatico. Per generare token di accesso per l' machine-to-machine autorizzazione con ambiti OAuth 2.0, puoi aggiungere un client di app che genera concessioni di credenziali client.

Argomenti correlati

- Autorizzazione Scopes, M2M e API con server di risorse

Interfaccia utente ospitata

Quando non desideri creare un'interfaccia utente, puoi presentare agli utenti un'interfaccia utente personalizzata ospitata da Amazon Cognito. L'interfaccia utente ospitata è un insieme di pagine web per la registrazione, l'accesso, l'autenticazione a più fattori (MFA) e la reimpostazione della password. Puoi aggiungere l'interfaccia utente ospitata al tuo dominio esistente o utilizzare un identificatore di prefisso in un sottodominio. AWS

Argomenti correlati

- Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione
- Configurazione di un dominio di bacino d'utenza

Sicurezza

Gli utenti locali possono fornire un fattore di autenticazione aggiuntivo con un codice di un messaggio SMS o un'app che genera codici di autenticazione a più fattori (MFA). Puoi creare meccanismi per configurare ed elaborare l'autenticazione MFA nell'app oppure lasciare che sia l'interfaccia utente ospitata a gestirla. I pool di utenti Amazon Cognito possono ignorare l'autenticazione MFA quando gli utenti accedono da dispositivi affidabili.

Se inizialmente non desideri richiedere l'autenticazione MFA agli utenti, puoi richiederla in modo condizionale. Con funzionalità di sicurezza avanzate, Amazon Cognito è in grado di rilevare potenziali attività dannose e richiedere all'utente di configurare l'autenticazione MFA o bloccare l'accesso.

Se il traffico di rete verso il tuo pool di utenti potrebbe essere dannoso, puoi monitorarlo e intervenire con AWS WAF gli ACL web.

Argomenti correlati

- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Aggiunta di sicurezza avanzata a un bacino d'utenza
- Associazione di un ACL Web a un pool di utenti AWS WAF

Esperienza utente personalizzata

Nella maggior parte delle fasi di registrazione, accesso o aggiornamento profilo di un utente, puoi personalizzare il modo in cui Amazon Cognito gestisce la richiesta. Con i trigger Lambda, puoi modificare un token ID o rifiutare una richiesta di iscrizione in base a condizioni personalizzate. Puoi creare un flusso di autenticazione personalizzato.

Puoi caricare CSS e logo personalizzati affinché l'interfaccia utente ospitata abbia un aspetto familiare per gli utenti.

Argomenti correlati

- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Trigger Lambda di richieste di autenticazione personalizzate
- Personalizzazione delle pagine Web di registrazione e accesso integrate

Monitoraggio e analisi

I pool di utenti Amazon Cognito registrano le richieste API, incluse le richieste all'interfaccia utente ospitata, in AWS CloudTrail. Puoi rivedere le metriche delle prestazioni in Amazon CloudWatch Logs, inviare log personalizzati con trigger CloudWatch Lambda e monitorare il volume delle richieste API nella console Service Quotas.

Puoi anche registrare i dati del dispositivo e della sessione dalle richieste API in una campagna Amazon Pinpoint. Con Amazon Pinpoint, puoi inviare notifiche push dall'app in base all'analisi dell'attività degli utenti.

Argomenti correlati

- Registrazione delle chiamate all'API Amazon Cognito con AWS CloudTrail
- Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas
- Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Integrazione dei pool di identità di Amazon Cognito

L'altra metà di Amazon Cognito è costituita da pool di identità. I pool di identità forniscono credenziali che autorizzano e monitorano le richieste API inviate dai Servizi AWS tuoi utenti, ad esempio ad Amazon DynamoDB o Amazon S3. Puoi creare policy di accesso basate sull'identità che proteggono i dati in base alla classificazione degli utenti nel pool di utenti. I pool di identità possono anche accettare token e asserzioni SAML 2.0 da un'ampia gamma di provider di identità, a prescindere dall'autenticazione del pool di utenti.

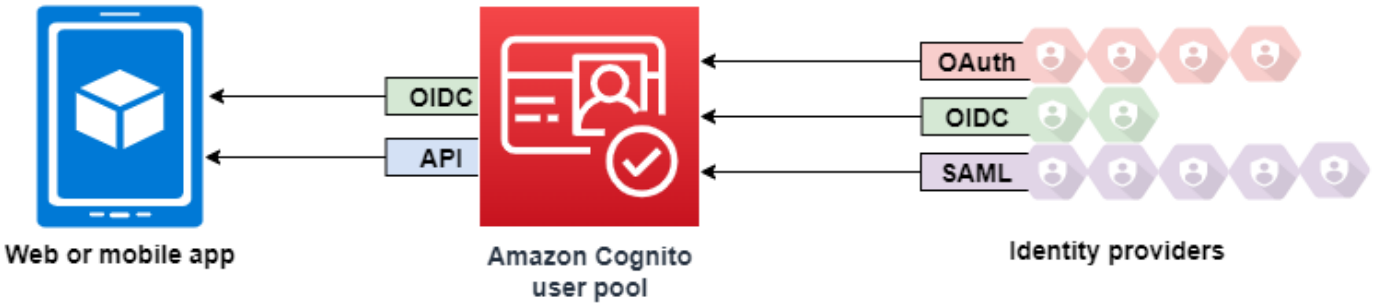
Argomenti correlati

- Accesso Servizi AWS tramite un pool di identità dopo l'accesso
- Pool di identità di Amazon Cognito

Autenticazione con un bacino d'utenza

Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs

Una volta completata l'autenticazione, Amazon Cognito restituisce i token del bacino d'utenza all'App. Puoi utilizzare i token per concedere agli utenti l'accesso alle risorse lato server o ad Amazon API Gateway. In alternativa, puoi scambiarli con credenziali per accedere ad altri servizi. AWS AWS



Il trattamento e la gestione dei token del bacino d'utenza per l'App Web o per dispositivi mobili avvengono lato client tramite gli SDK di Amazon Cognito. In modo analogo, Mobile SDK for iOS e Mobile SDK for Android aggiornano automaticamente i token ID e di accesso se esiste un token di aggiornamento valido (non scaduto) e i token ID e di accesso hanno un validità residua di almeno 5 minuti. Per informazioni sugli SDK e sul codice di esempio per Android e iOS JavaScript, consulta gli SDK del pool di utenti di Amazon Cognito.

Dopo che l'utente dell'app ha effettuato l'accesso, Amazon Cognito crea una sessione e restituisce un token ID, un token di accesso e un token di aggiornamento per l'utente autenticato.

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;
}];
```

Argomenti

- Flusso di autenticazione del bacino d'utenza
- Client dell'app pool di utenti
- Utilizzo dei dispositivi utente nel pool di utenti

Flusso di autenticazione del bacino d'utenza

Amazon Cognito include diversi metodi per autenticare gli utenti. Tutti i pool di utenti, a prescindere che si disponga di un dominio, possono autenticare gli utenti nell'API dei pool di utenti. Se aggiungi un dominio al pool di utenti, puoi utilizzare gli endpoint del pool di utenti. L'API dei pool di utenti supporta una vasta gamma di modelli di autorizzazione e flussi di richiesta per le richieste API.

Per verificare l'identità degli utenti, Amazon Cognito supporta i flussi di autenticazione che integrano nuovi tipi di richieste, in aggiunta alle password. L'autenticazione in Amazon Cognito richiede in genere l'implementazione di due operazioni API nel seguente ordine:

Public authentication

1. InitiateAuth
2. RespondToAuthChallenge

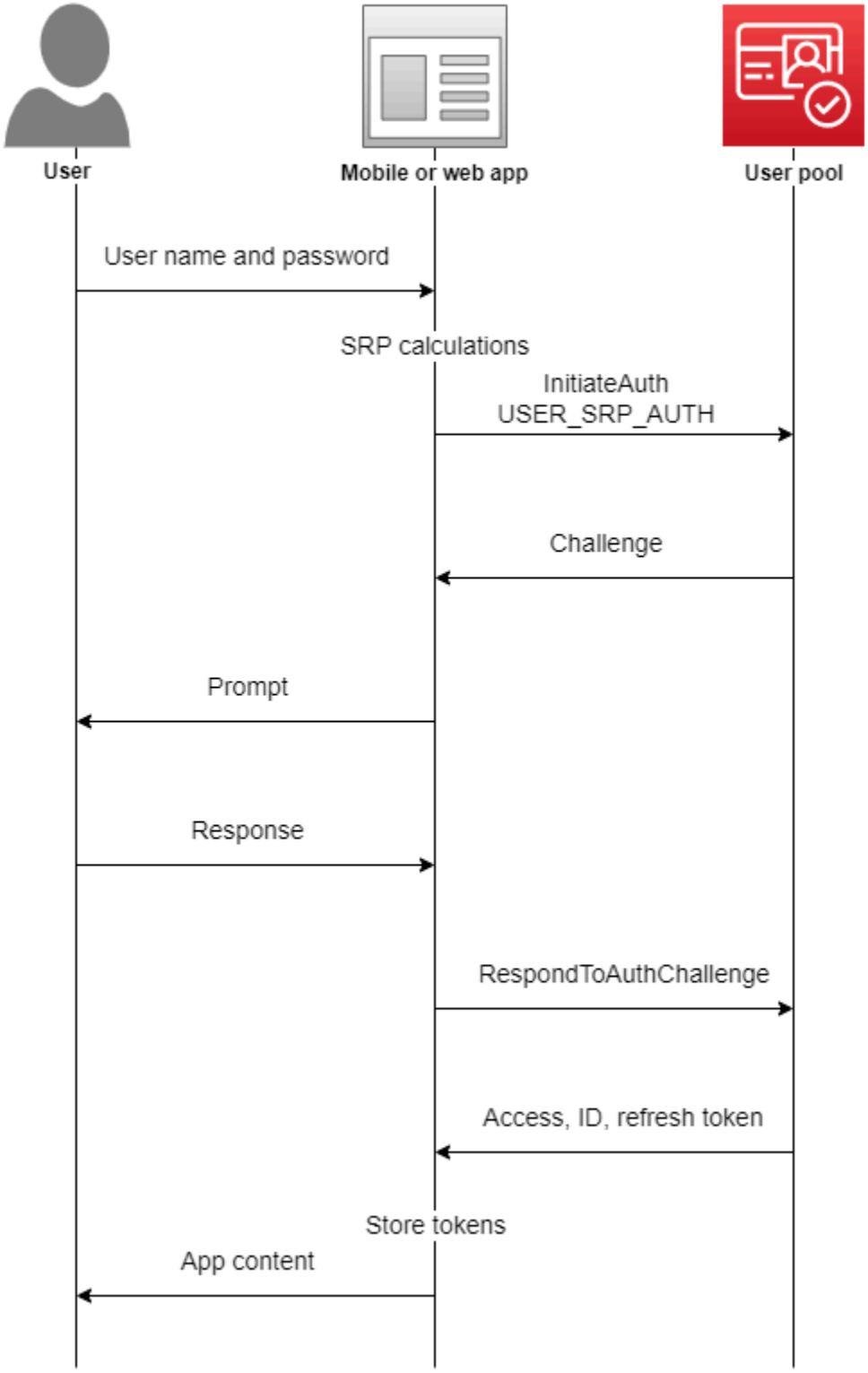
   `InitiateAuth` e `RespondToAuthChallenge` sono API non autenticate da utilizzare con client di app pubbliche lato client.

Server-side authentication

1. AdminInitiateAuth
2. AdminRespondToAuthChallenge

   `AdminInitiateAuth` e `AdminRespondToAuthChallenge` richiedono credenziali IAM e sono adatte per client di app riservate lato server.

L'autenticazione di un utente viene effettuata rispondendo a richieste successive fino a che l'autenticazione non riesce o Amazon Cognito emette token per l'utente. Puoi ripetere questi passaggi con Amazon Cognito, in un processo che include richieste di autenticazione di tipo diverso, per supportare qualsiasi flusso di autenticazione personalizzato.


User
Mobile or web app
User pool
User name and password
SRP calculations
InitiateAuth
USER_SRP_AUTH
Challenge
Prompt
Response
RespondToAuthChallenge
Access, ID, refresh token
Store tokens
App content

In genere, l'app genera un prompt per raccogliere informazioni dall'utente e le invia ad Amazon Cognito in una richiesta API. Considerare flusso `InitiateAuth` in un pool di utenti in cui l'utente è stato configurato con l'autenticazione a più fattori (MFA).

1. L'app richiede all'utente il nome utente e password.
2. Il nome utente e la password vengono inclusi come parametri in `InitiateAuth`.
3. Amazon Cognito restituisce una verifica `SMS_MFA` e un identificatore di sessione.
4. L'app richiede all'utente di inserire il codice MFA dal telefono.
5. Includere il codice e l'identificatore di sessione nella richiesta `RespondToAuthChallenge`.

A seconda delle caratteristiche del pool di utenti, è possibile dover rispondere a diverse verifiche `InitiateAuth` prima che l'app recuperi i token da Amazon Cognito. Amazon Cognito include una stringa di sessione nella risposta a ciascuna richiesta. Per combinare le richieste API in un flusso di autenticazione, includere la stringa di sessione della risposta alla richiesta precedente in ogni richiesta successiva. Per impostazione predefinita, gli utenti hanno a disposizione tre minuti per completare ogni verifica prima della scadenza della stringa della sessione. Per modificare questo periodo, modificare il client dell'app Authentication flow session duration (Durata della sessione del flusso di autenticazione). La procedura seguente descrive come modificare questa impostazione nella configurazione del client dell'app.

> **Note**
>
> Le impostazioni Durata della sessione del flusso di autenticazione si applicano all'autenticazione con l'API dei pool di utenti Amazon Cognito. L'interfaccia utente ospitata di Amazon Cognito imposta la durata della sessione su 3 minuti per l'autenticazione a più fattori e su 8 minuti per i codici di reimpostazione della password.

Amazon Cognito console

Configurare la durata della sessione del flusso di autenticazione dell'app client (AWS Management Console)

1. Nella scheda App integration (Integrazione app) del pool di utenti, seleziona il nome del client dell'app nel container App clients and analytics (Client di app e analisi dei dati).
2. Scegli Modifica nel container Informazioni sul client dell'app.

3. Cambia il valore dell'opzione Authentication flow session duration (Durata della sessione del flusso di autenticazione) impostando la durata di validità desiderata, espressa in minuti, per i codici dell'autenticazione MFA con SMS. Ciò modifica anche il tempo che ogni utente ha a disposizione per completare qualsiasi richiesta di autenticazione nel client dell'app.
4. Seleziona Salvataggio delle modifiche.

Amazon Cognito API

Configurare la durata della sessione del flusso di autenticazione del client dell'app (API Amazon Cognito)

1. Prepara una richiesta `UpdateUserPoolClient` con le impostazioni di un pool di utenti esistente in base a una richiesta `DescribeUserPoolClient`. La richiesta `UpdateUserPoolClient` deve includere tutte le proprietà client dell'app esistenti.
2. Cambia il valore del parametro `AuthSessionValidity` impostando la durata di validità desiderata, espressa in minuti, per i codici dell'autenticazione MFA con SMS. Ciò modifica anche il tempo che ogni utente ha a disposizione per completare qualsiasi richiesta di autenticazione nel client dell'app.

Per ulteriori informazioni sui client di app, consulta Client dell'app pool di utenti.

Puoi utilizzare i AWS Lambda trigger per personalizzare il modo in cui gli utenti effettuano l'autenticazione. Questi trigger generano e verificano le proprie richieste come parte del flusso di autenticazione.

Inoltre puoi utilizzare il flusso di autenticazione amministratore per server back-end protetti. Con il flusso di autenticazione per la migrazione degli utenti, tale migrazione è possibile senza richiedere agli utenti di reimpostare le password.

Comportamento di blocco di Amazon Cognito in caso di tentativi di accesso non riusciti

Dopo cinque tentativi di accesso non autenticato o autenticato mediante IAM non riusciti, Amazon Cognito blocca l'utente per un secondo. La durata del blocco quindi raddoppia dopo ogni ulteriore tentativo non riuscito, fino a un massimo di circa 15 minuti. I tentativi effettuati durante un periodo di blocco generano un'eccezione `Password attempts exceeded` e non influiscono sulla durata dei periodi di blocco successivi. Per un numero cumulativo di tentativi di accesso non riusciti n, ad esclusione delle eccezioni `Password attempts exceeded`, Amazon Cognito blocca l'utente per 2^(n-5) secondi. Per ripristinare lo stato iniziale n=0 del blocco, l'utente deve effettuare un accesso

riuscito dopo la scadenza del periodo di blocco oppure non deve iniziare alcun tentativo di accesso per 15 minuti consecutivi in qualsiasi momento dopo il blocco. Questo comportamento è soggetto a modifiche. Questo comportamento non si applica alle sfide personalizzate a meno che non eseguano anche l'autenticazione basata su password.

Argomenti

- Flusso di autenticazione lato client
- Flusso di autenticazione lato server
- Flusso di autenticazione personalizzato
- Flusso di autenticazione integrato e sfide
- Flusso di autenticazione personalizzato e sfide
- Utilizzare la verifica della password SRP nel flusso di autenticazione personalizzato
- Flusso di autenticazione amministratore
- Flusso di autenticazione per la migrazione degli utenti

Flusso di autenticazione lato client

Il seguente processo funziona per le app lato client dell'utente create con AWS Amplify o SDK AWS.

1. L'utente inserisce il nome utente e la password nell'app.
2. L'app chiama l'operazione `InitiateAuth` con il nome utente e i dettagli SRP (Secure Remote Password) dell'utente.

   Questa operazione API restituisce i parametri di autenticazione.

   > **Note**
   >
   > L'app genera i dettagli SRP con le funzionalità SRP di Amazon Cognito integrate negli SDK AWS .

3. L'app chiama l'operazione `RespondToAuthChallenge`. Se la chiamata va a buon fine, Amazon Cognito restituisce i token dell'utente e il flusso di autenticazione è completo.

   Se Amazon Cognito richiede un ulteriore fattore di autenticazione, la chiamata a `RespondToAuthChallenge` non restituisce token. Invece, la chiamata restituisce una sessione.

4. Se `RespondToAuthChallenge` restituisce una sessione, l'app chiama di nuovo `RespondToAuthChallenge`, questa volta con la sessione e la risposta alla richiesta di identificazione (ad esempio, il codice MFA).

Flusso di autenticazione lato server

Se non disponi di un'app utente, ma usi un backend sicuro Java, Ruby o Node.js o un'app lato server, puoi usare l'API autenticata lato server per il bacino d'utenza di Amazon Cognito.

Per le app lato server, l'autenticazione dei bacini d'utenza è analoga a quella per le app lato client, con le seguenti eccezioni:

- L'app lato server chiama l'operazione API `AdminInitiateAuth` (invece di `InitiateAuth`). Questa operazione richiede AWS credenziali con autorizzazioni che includono e. `cognito-idp:AdminInitiateAuth` `cognito-idp:AdminRespondToAuthChallenge` Questa operazione restituisce i parametri di autenticazione richiesti.
- Una volta che l'app lato server dispone dei parametri di autenticazione, chiama l'operazione API `AdminRespondToAuthChallenge` (anziché `RespondToAuthChallenge`). L'operazione `AdminRespondToAuthChallenge` API ha esito positivo solo quando si forniscono le credenziali. AWS

Per ulteriori informazioni sulla firma delle richieste API di Amazon Cognito con AWS credenziali, consulta il processo di firma Signature versione 4 nella AWS Guida generale.

Le operazioni `AdminInitiateAuth` e le `AdminRespondToAuthChallenge` API non possono accettare le credenziali username-and-password utente per l'accesso da amministratore, a meno che tu non le abiliti esplicitamente a farlo in uno dei seguenti modi:

- Include `ALLOW_ADMIN_USER_PASSWORD_AUTH` (precedentemente noto come `ADMIN_NO_SRP_AUTH`) nel parametro `ExplicitAuthFlow` quando chiami `CreateUserPoolClient` o `UpdateUserPoolClient`.
- Aggiungi `ALLOW_ADMIN_USER_PASSWORD_AUTH` all'elenco dei flussi di autenticazione per il client dell'app. Configura i client di app nella scheda App integration (Integrazione app) del bacino d'utenza, sotto App clients and analytics (Client di app e analisi). Per ulteriori informazioni, consulta Client dell'app pool di utenti.

Flusso di autenticazione personalizzato

I pool di utenti di Amazon Cognito consentono inoltre di utilizzare flussi di autenticazione personalizzati, che possono aiutarti a creare un modello di autenticazione basato su sfida/risposta utilizzando i trigger. AWS Lambda

> **Note**
>
> Non puoi utilizzare le funzionalità di sicurezza avanzate per le credenziali compromesse e l'autenticazione adattiva con flussi di autenticazione personalizzati. Per ulteriori informazioni, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

Il flusso di autenticazione personalizzato permette cicli personalizzati di richieste e risposte per soddisfare diversi requisiti. Il flusso comincia con una chiamata all'operazione API `InitiateAuth` che indica il tipo di autenticazione da utilizzare e fornisce qualsiasi parametro di autenticazione iniziale. Amazon Cognito risponde alla chiamata `InitiateAuth` con uno dei seguenti tipi di informazione:

- Una sfida all'utente, insieme a una sessione e parametri.
- Un errore se l'utente non si autentica correttamente.
- Token ID, di accesso e di aggiornamento, se i parametri forniti nella chiamata `InitiateAuth` sono sufficienti per l'accesso dell'utente. (In genere l'utente o l'app devono prima rispondere a una sfida, ma è il tuo codice personalizzato a decidere se è il caso.)

Se Amazon Cognito risponde alla chiamata `InitiateAuth` con una sfida, l'app raccoglierà più input e chiamerà l'operazione `RespondToAuthChallenge`. Questa chiamata fornisce le risposte alla sfida e restituisce la sessione. Amazon Cognito risponde alla chiamata `RespondToAuthChallenge` in modo simile alla chiamata `InitiateAuth`. Se l'utente ha effettuato l'accesso, Amazon Cognito fornisce i token, se non ha effettuato l'accesso, Amazon Cognito restituisce un'altra sfida o un errore. Se Amazon Cognito restituisce un'altra sfida, la sequenza si ripete: l'app chiama `RespondToAuthChallenge` fino a quando l'utente non riesce a effettuare l'accesso o non viene restituito un errore. Per ulteriori dettagli sulle operazioni `InitiateAuth` e API `RespondToAuthChallenge`, consulta la documentazione API.

Flusso di autenticazione integrato e sfide

Amazon Cognito contiene valori `AuthFlow` e `ChallengeName` integrati affinché un flusso di autenticazione standard possa convalidare nome utente e password tramite il protocollo Secure Remote Password (SRP). Gli AWS SDK offrono un supporto integrato per questi flussi con Amazon Cognito.

Il flusso inizia inviando `USER_SRP_AUTH` come `AuthFlow` a `InitiateAuth`. Dovrai inviare anche i valori `USERNAME` e `SRP_A` in `AuthParameters`. Se la chiamata `InitiateAuth` ha esito positivo, la risposta include `PASSWORD_VERIFIER` come `ChallengeName` e `SRP_B` nei parametri di sfida. L'app chiama quindi `RespondToAuthChallenge` con `PASSWORD_VERIFIER` `ChallengeName` e i parametri necessari in `ChallengeResponses`. Se la chiamata a `RespondToAuthChallenge` ha esito positivo e l'utente effettua l'accesso, Amazon Cognito emette token. Se hai attivato l'autenticazione a più fattori (MFA) per l'utente, Amazon Cognito restituisce il `ChallengeName` di `SMS_MFA`. L'app può fornire il codice necessario tramite un'altra chiamata a `RespondToAuthChallenge`.

Flusso di autenticazione personalizzato e sfide

Un'app è in grado di avviare un flusso di autenticazione chiamando `InitiateAuth` con `CUSTOM_AUTH` come `Authflow`. Nel caso di un flusso di autenticazione personalizzato, tre trigger Lambda controllano le sfide e la verifica delle risposte.

- Il trigger Lambda `DefineAuthChallenge` usa come input una matrice di sessioni di sfide e risposte precedenti. Quindi genera il nome della sfida successiva e i booleani che indicano se l'utente è autenticato e deve ricevere i token o meno. Questo trigger Lambda è una macchina a stati che controlla il percorso dell'utente attraverso le richieste.
- Il trigger Lambda `CreateAuthChallenge` utilizza il nome di una sfida come input e genera la sfida e i parametri per valutare la risposta. Quando `DefineAuthChallenge` restituisce `CUSTOM_CHALLENGE` come sfida sucessiva, il flusso di autenticazione chiama `CreateAuthChallenge`. Il trigger Lambda `CreateAuthChallenge` passa il tipo di sfida successiva nel parametro di metadati della richiesta.
- La funzione Lambda `VerifyAuthChallengeResponse` valuta la risposta e restituisce un valore booleano per indicare se la risposta era valida.

Un flusso di autenticazione personalizzato può anche utilizzare una combinazione di sfide incorporate, come la verifica della password SRP e MFA tramite SMS. Può utilizzare sfide personalizzate come CAPTCHA o domande segrete.

Utilizzare la verifica della password SRP nel flusso di autenticazione personalizzato

Se intendi includere la verifica SRP in un flusso di autenticazione personalizzato, è necessario partire da essa.

- Per avviare la verifica della password SRP in un flusso personalizzato, l'app chiama `InitiateAuth` con `CUSTOM_AUTH` come `Authflow`. Nella mappa `AuthParameters`, la richiesta dalla tua app include `SRP_A:` (il valore SRP A) e `CHALLENGE_NAME: SRP_A`.
- Il flusso `CUSTOM_AUTH` richiama il trigger Lambda `DefineAuthChallenge` con una sessione iniziale di `challengeName: SRP_A` e `challengeResult: true`. La tua funzione Lambda risponde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false` e `failAuthentication: false`.
- L'app deve quindi chiamare `RespondToAuthChallenge` con `challengeName: PASSWORD_VERIFIER` e gli altri parametri necessari per SRP nella mappa `challengeResponses`.
- Se Amazon Cognito verifica la password, `RespondToAuthChallenge` richiama il trigger Lambda `DefineAuthChallenge` con una seconda sessione di `challengeName: PASSWORD_VERIFIER` e `challengeResult: true`. A quel punto, il trigger `DefineAuthChallenge` Lambda risponde con `challengeName: CUSTOM_CHALLENGE` per avviare la richiesta personalizzata.
- Se MFA è abilitato per un utente, dopo che Amazon Cognito verifica la password, all'utente viene richiesto di configurare o accedere con MFA.

> **Note**
>
> La pagina Web di accesso ospitata di Amazon Cognito non può attivare i Trigger Lambda di richieste di autenticazione personalizzate.

Per ulteriori informazioni sui trigger lambda, compreso il codice di esempio, consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Flusso di autenticazione amministratore

La best practice per l'autenticazione consiste nell'utilizzare le operazioni API descritte in Flusso di autenticazione personalizzato con SRP per la verifica della password. Gli AWS SDK utilizzano questo approccio e questo approccio li aiuta a utilizzare SRP. Tuttavia, per evitare i calcoli di SRP, è disponibile un set alternativo di operazioni API di amministrazione disponibili per l'utilizzo su server di back-end protetti. Per queste implementazioni di amministrazione back-end, utilizza `AdminInitiateAuth` al posto di `InitiateAuth`. Inoltre, utilizza `AdminRespondToAuthChallenge` al posto di `RespondToAuthChallenge`. Grazie alla possibilità di inviare la password come testo normale, non è necessario eseguire calcoli SRP quando si utilizzano queste operazioni. Ecco un esempio:

```
AdminInitiateAuth Request {
    "AuthFlow":"ADMIN_USER_PASSWORD_AUTH",
    "AuthParameters":{
        "USERNAME":"<username>",
        "PASSWORD":"<password>"
        },
    "ClientId":"<clientId>",
    "UserPoolId":"<userPoolId>"
}
```

Queste operazioni API di autenticazione di amministrazione richiedono credenziali per gli sviluppatori e utilizzano il processo di firma di AWS Signature Version 4 (SigV4). Queste API sono disponibili negli SDK AWS standard, come Node.js, molto utile per le funzioni Lambda. Per utilizzare queste operazioni e fare loro accettare le password in testo normale, è necessario attivarle per l'app nella console. In alternativa, puoi passare `ADMIN_USER_PASSWORD_AUTH` per il parametro `ExplicitAuthFlow` nelle chiamate a `CreateUserPoolClient` o `UpdateUserPoolClient`. Le operazioni `InitiateAuth` e `RespondToAuthChallenge` non accettano `ADMIN_USER_PASSWORD_AUTH` `AuthFlow`.

Nella risposta di `AdminInitiateAuth` `ChallengeParameters`, l'attributo `USER_ID_FOR_SRP`, se presente, contiene il nome utente reale dell'utente e non un alias (come l'indirizzo e-mail o il numero di telefono). Nella chiamata a `AdminRespondToAuthChallenge`, in `ChallengeResponses`, è necessario passare questo nome utente nel parametro `USERNAME`.

> **Note**
>
> Dato che le implementazioni di amministrazione di back-end utilizzano il flusso di autenticazione di amministrazione, il flusso non supporta il tracciamento dei dispositivi. Quando abiliti il tracciamento dei dispositivi, l'autenticazione di amministrazione avviene correttamente, ma le chiamate per aggiornare i token di accesso non andranno a buon fine.

Flusso di autenticazione per la migrazione degli utenti

Il trigger Lambda per la migrazione degli utenti aiuta a migrare gli utenti da un sistema legacy di gestione degli utenti al bacino d'utenza. Se scegli il flusso di autenticazione `USER_PASSWORD_AUTH`, gli utenti non dovranno reimpostare le password durante la migrazione. Questo flusso invia le password degli utenti al servizio tramite una connessione SSL crittografata durante l'autenticazione.

Una volta completata la migrazione di tutti gli utenti, passa i flussi al flusso SRP più sicuro. Il flusso SRP non invia le password sulla rete.

Per ulteriori informazioni sui trigger Lambda, consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Per ulteriori informazioni sulla migrazione degli utenti con un trigger Lambda, consulta Importazione di utenti in bacini d'utenza con un trigger Lambda di migrazione utenti.

Client dell'app pool di utenti

Un client dell'app del pool di utenti è una configurazione all'interno di un pool di utenti che interagisce con un'applicazione per dispositivi mobili o Web che esegue l'autenticazione con Amazon Cognito. I client di app possono chiamare operazioni API autenticate e non autenticate, nonché leggere o modificare alcuni o tutti gli attributi degli utenti. L'app deve identificarsi con il client dell'app durante le operazioni di registrazione, accesso e gestione delle password dimenticate. Queste richieste API devono includere identificazione automatica con un ID del client dell'app e l'autorizzazione con un segreto client opzionale. È responsabilità dell'utente proteggere eventuali ID o segreti del client dell'app in modo che solo le app client autorizzate possano chiamare queste operazioni non autenticate. Inoltre, se configuri la tua app per firmare richieste API autenticate con AWS credenziali, devi proteggere le tue credenziali dall'ispezione da parte degli utenti.

È possibile creare più client app per un bacino d'utenza. Un client dell'app potrebbe essere collegato alla piattaforma di codice di un'app o a un tenant separato nel pool di utenti. Ad esempio, puoi creare un'app per un'applicazione lato server e un'app Android diversa. Ogni app ha il proprio ID client.

Tipi di client di app

Quando crei un client dell'app in Amazon Cognito, puoi precompilare le opzioni in base ai tipi di client OAuth standard client pubblico e client riservato. Configurare un client riservato con un client secret. Per ulteriori informazioni sui tipi di client, vedere IETF RFC 6749 #2.1.

Client pubblico

Un client pubblico viene eseguito in un browser o su un dispositivo mobile. Poiché non dispone di risorse affidabili sul lato server, non ha un client secret.

Client riservato

Un client riservato dispone di risorse lato server che possono essere attendibili e di un client secret per operazioni API non autenticate L'app potrebbe essere eseguita come daemon o script di shell sul server back-end.

Client secret

Un client secret, o client password, è una stringa fissa che l'app deve utilizzare in tutte le richieste API che invia al client dell'app. Il client dell'app deve avere un client secret per eseguire la concessione di `client_credentials`. Per ulteriori informazioni, consulta IETF RFC 6749 #2.3.1.

Non puoi modificare il client secret una volta che hai creato un'app. Puoi creare una nuova app con un nuovo client secret se desideri ruotare la chiave privata che stai utilizzando. Puoi anche eliminare un'app per bloccare l'accesso da app che usano l'ID client di quell'app.

È possibile utilizzare un client riservato e un client secret con un'app pubblica. Usa un CloudFront proxy Amazon per aggiungere un proxy `SECRET_HASH` in transito. Per ulteriori informazioni, consulta Proteggere i client pubblici per Amazon Cognito utilizzando un CloudFront proxy Amazon sul AWS blog.

Token web JSON

I client dell'app Amazon Cognito possono emettere token web JSON (JWT) dei seguenti tipi.

Token di identità (ID)

Una dichiarazione verificabile attestante che l'utente è autenticato dal pool di utenti. OpenID Connect (OIDC) ha aggiunto la specifica del token ID agli standard dei token di accesso e aggiornamento definiti da OAuth 2.0. Il token ID contiene informazioni sull'identità, come gli attributi utente, che l'app può utilizzare per creare un profilo utente e fornire risorse. Per ulteriori informazioni, consulta Utilizzo di token ID.

Token di accesso

Una dichiarazione verificabile dei diritti di accesso dell'utente. Il token di accesso contiene gli ambiti, una funzionalità di OIDC e OAuth 2.0. L'app può presentare gli ambiti delle risorse di back-end e dimostrare che il pool di utenti ha autorizzato un utente o una macchina ad accedere ai dati da un'API o ai propri dati utente. Un token di accesso con ambiti personalizzati, spesso ottenuto tramite una concessione di credenziali client M2M, autorizza l'accesso a un server di risorse. Per ulteriori informazioni, consulta Utilizzo del token di accesso.

Token di aggiornamento

Una dichiarazione crittografata di autenticazione iniziale che l'app può presentare al pool di utenti quando i token dell'utente scadono. Una richiesta refresh-token restituisce token di accesso e ID nuovi e non scaduti. Per ulteriori informazioni, consulta Utilizzo del token di aggiornamento.

Puoi impostare la scadenza di questi token per ogni client dell'app dalla scheda Integrazione app del tuo pool di utenti nella console Amazon Cognito.

Termini del client dell'app

I termini seguenti sono proprietà disponibili dei client di app nella console Amazon Cognito.

URL di callback consentiti

Un URL di callback indica dove l'utente deve essere reindirizzato dopo aver effettuato correttamente l'accesso. Scegli almeno un URL di callback. L'URL di callback deve:

- Deve essere un URI assoluto.
- Deve essere preregistrato con un client.
- Non deve includere un componente di frammento.

Consulta OAuth 2.0 - redirection endpoint (OAuth 2.0 - Endpoint di reindirizzamento).

Amazon Cognito richiede HTTPS su HTTP tranne che per `http://localhost` a solo scopo di test.

Sono supportati anche URL di callback come `myapp://example`.

URL di disconnessione consentiti

Un URL di disconnessione indica dove l'utente deve essere reindirizzato dopo la disconnessione.

Autorizzazioni di lettura e scrittura dell'attributo

Il tuo pool di utenti potrebbe avere molti clienti, ognuno con il proprio client di app e IdPs. Puoi configurare il client dell'app in modo che abbia accesso in lettura e scrittura solo agli attributi utente pertinenti all'app. In casi come l'autorizzazione machine-to-machine (M2M), puoi concedere l'accesso a nessuno dei tuoi attributi utente.

Considerazioni sulla configurazione delle autorizzazioni di lettura e scrittura degli attributi

- Quando crei un client per app e non personalizzi le autorizzazioni di lettura e scrittura degli attributi, Amazon Cognito concede le autorizzazioni di lettura e scrittura a tutti gli attributi del pool di utenti.
- Puoi concedere l'accesso in scrittura ad attributi personalizzati non modificabili. L'app client può scrivere un valore in un attributo non modificabile solo quando crei o registri un utente. Dopodiché, non puoi scrivere valori in alcun attributo personalizzato non modificabile per l'utente.
- I client dell'app devono avere accesso in scrittura agli attributi richiesti nel pool di utenti. La console Amazon Cognito imposta automaticamente gli attributi richiesti come scrivibili.
- Non puoi consentire a un client dell'app di avere accesso in scrittura a `email_verified` o a `phone_number_verified`. Un amministratore del pool di utenti può modificare questi valori. Un utente può modificare il valore di questi attributi solo tramite la verifica degli attributi.

Flusso di autenticazione

I metodi di accesso consentiti dal client dell'app. L'app può supportare l'autenticazione con nome utente e password, Secure Remote Password (SRP), l'autenticazione personalizzata con trigger Lambda e l'aggiornamento dei token. Come migliore pratica di sicurezza, utilizza l'autenticazione SRP come metodo di accesso principale. L'interfaccia utente ospitata accede automaticamente agli utenti con SRP.

Ambiti personalizzati

Un ambito personalizzato è quello che viene definito per il proprio server di risorse nella scheda Resource Servers (Server di Risorse). *Il formato è/scope. resource-server-identifier* Per informazioni, consulta Autorizzazione Scopes, M2M e API con server di risorse.

URI di reindirizzamento predefinito

Sostituisce il `redirect_uri` parametro nelle richieste di autenticazione per gli utenti con terze parti. IdPs Configura questa impostazione del client dell'app con il `DefaultRedirectURI` parametro di una richiesta CreateUserPoolClientto UpdateUserPoolClientAPI. Questo URL deve inoltre essere un membro del client `CallbackURLs` for your app. Amazon Cognito reindirizza le sessioni autenticate a questo URL quando:

1. Al client dell'app è assegnato un provider di identità e sono definiti più URL di callback. Il pool di utenti reindirizza le richieste di autenticazione al server di autorizzazione all'URI di reindirizzamento predefinito quando non includono un parametro. `redirect_uri`
2. Il client dell'app ha un provider di identità assegnato e un URL di callback definito. In questo scenario non è necessario definire un URL di callback predefinito. Le richieste che non includono un `redirect_uri` parametro reindirizzano all'unico URL di callback disponibile.

Provider di identità

Puoi scegliere alcuni o tutti i provider di identità esterni del tuo pool di utenti (IdPs) per autenticare i tuoi utenti. Il client di app può inoltre autenticare solo gli utenti locali del pool di utenti. Quando si aggiunge un provider di identità al client di app, è possibile generare link di autorizzazione per l'IdP e visualizzarli nella pagina di accesso dell'interfaccia utente ospitata. È possibile assegnarne più di uno IdPs, ma è necessario assegnarne almeno uno. Per ulteriori informazioni sull'utilizzo di dispositivi esterni IdPs, vedere. Aggiunta di un accesso al bacino d'utenza tramite terze parti

Ambiti OpenID Connect

Scegli uno o più dei seguenti ambiti `OAuth` per specificare i privilegi di accesso che possono essere richiesti per i token d'accesso.

- L'ambito `openid` dichiara che desideri recuperare un token ID e l'ID univoco di un utente. Richiede inoltre tutti o alcuni attributi utente, a seconda degli ambiti aggiuntivi nella richiesta. Amazon Cognito non restituisce un token ID a meno che non venga richiesto l'ambito `openid`. L'ambito `openid` autorizza le attestazioni dei token ID strutturali come la scadenza e l'ID chiave e determina gli attributi utente che ricevi in una risposta da Endpoint UserInfo.

  - Quando `openid` è l'unico ambito richiesto, Amazon Cognito popola il token ID con tutti gli attributi utente che il client dell'app corrente è in grado di leggere. La risposta `userInfo` a un token di accesso con questo ambito restituisce tutti gli attributi utente.
  - Quando richiedi `openid` con altri ambiti come `phone`, `email` o `profile`, il token ID e `userInfo` restituiscono l'ID univoco dell'utente e gli attributi definiti dagli ambiti aggiuntivi.
- L'ambito `phone` permette l'accesso alle richieste `phone_number` e `phone_number_verified`. Questo ambito può essere richiesto solo con l'ambito `openid`.
- L'ambito `email` permette l'accesso alle richieste `email` e `email_verified`. Questo ambito può essere richiesto solo con l'ambito `openid`.
- L'`aws.cognito.signin.user.admin`ambito consente l'accesso alle operazioni API dei pool di utenti di Amazon Cognito che richiedono token di accesso, come e. UpdateUserAttributesVerifyUserAttribute
- L'ambito `profile` permette l'accesso a tutti gli attributi dell'utente negli ID token che sono leggibili dal client. Questo ambito può essere richiesto solo con l'ambito `openid`.

Per ulteriori informazioni sugli ambiti, consulta la lista degli ambiti OIDC standard.

Tipi di concessione OAuth

Una concessione OAuth è un metodo di autenticazione che recupera i token del pool di utenti. Amazon Cognito supporta i seguenti tipi di concessioni. Per integrare queste concessioni OAuth nella tua app, devi aggiungere un dominio al tuo pool di utenti.

Concessione codice autorizzazione

La concessione del codice di autorizzazione genera un codice che l'app può scambiare con i token del pool di utenti con Endpoint Token. Durante lo scambio di un codice di autorizzazione, l'app riceve token ID, di accesso e di aggiornamento. Questo flusso OAuth, come la concessione implicita, avviene nei browser degli utenti. La concessione di un codice di autorizzazione è la concessione più sicura offerta da Amazon Cognito, poiché i token non sono visibili nelle sessioni degli utenti. Invece, l'app genera la richiesta che restituisce i token e può memorizzarli nella cache in uno spazio di archiviazione protetto. Per ulteriori informazioni, vedere Codice di autorizzazione in IETF RFC 6749 #1 .3.1

> **Note**
>
> Come best practice di sicurezza nelle app per client pubblici, attiva solo il flusso OAuth di concessione del codice di autorizzazione e implementa Proof Key for Code Exchange

(PKCE) per limitare lo scambio di token. Con PKCE, un client può scambiare un codice di autorizzazione solo dopo aver fornito all'endpoint del token lo stesso segreto presentato nella richiesta di autenticazione originale. Per ulteriori informazioni, consulta IETF RFC 7636.

Implicit grant (Concessione implicita)

La concessione implicita fornisce un token di accesso e ID, ma non un token di aggiornamento, alla sessione del browser dell'utente direttamente da Endpoint Authorize. Una concessione implicita rimuove il requisito di una richiesta separata all'endpoint del token, ma non è compatibile con PKCE e non restituisce token di aggiornamento. Questa concessione supporta scenari di test e architetture di app che non consentono di completare la concessione di codici di autorizzazione. Per ulteriori informazioni, consulta Concessione implicita in IETF RFC 6749 #1.3.2. Puoi attivare sia la concessione del codice di autorizzazione sia la concessione implicita in un client dell'app e usare entrambe in base alle esigenze.

Concessione credenziali del client

La concessione delle credenziali del client è per le comunicazioni machine-to-machine (M2M). Il codice di autorizzazione e le concessioni implicite emettono token a utenti umani autenticati. Le credenziali del client concedono l'autorizzazione basata sull'ambito da un sistema non interattivo a un'API. L'app può richiedere le credenziali del client direttamente dall'endpoint del token e ricevere un token di accesso. Per ulteriori informazioni, consulta Credenziali del client in IETF RFC 6749 #1 .3.4. È possibile attivare la concessione di credenziali client solo nei client di app che dispongono di un client secret e che non supportano il codice di autorizzazione o le concessioni implicite.

> **Note**
>
> Poiché il flusso di credenziali del client non viene richiamato come un utente, questa concessione può solo aggiungere ambiti personalizzati per accedere ai token. Un ambito personalizzato è quello che viene definito per il proprio server di risorse. Gli ambiti predefiniti come `openid` e `profile` non si applicano agli utenti non umani.
> Poiché i token ID sono una convalida degli attributi utente, non sono rilevanti per la comunicazione M2M e una concessione delle credenziali del client non li rilascia. Per informazioni, consulta Autorizzazione Scopes, M2M e API con server di risorse.

Le credenziali del cliente garantiscono costi aggiuntivi alla fattura. AWS Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Creazione di un client dell'app

AWS Management Console

Come creare un client dell'app (console)

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Seleziona la scheda App integration (Integrazione App).
5. Alla voce App clients (Client dell'App), seleziona Create an App client (Crea un client dell'App).
6. Seleziona un App type (Tipo di App): Public client (Client pubblico), Confidential client (Client riservato), oppure Other (Altro).
7. Inserisci un nome del client dell'App.
8. Scegli Genera segreto del client per fare in modo che Amazon Cognito generi automaticamente un segreto del client. I segreti client sono generalmente associati ai client riservati.
9. Seleziona i Authentication flows (Flussi di autenticazione) che desideri per entrare nel tuo client App.
10. Configurazione di Authentication flow session duration (Durata della sessione del flusso di autenticazione). Questo è il tempo a disposizione degli utenti per completare ogni verifica di autenticazione prima della scadenza del token di sessione.
11. (Facoltativo) Se desideri configurare la scadenza dei token, completa la procedura seguente:
    a. Specifica la scadenza del token di aggiornamento del client dell'App. Il valore predefinito è 30 giorni. Puoi modificarlo con qualsiasi valore compreso tra 1 ora e 10 anni.
    b. Specifica la scadenza del token di accesso del client dell'App. Il valore predefinito è 1 ora. Puoi modificarlo con qualsiasi valore compreso tra 5 minuti e 24 ore.
    c. Specifica la scadenza del token ID del client dell'App. Il valore predefinito è 1 ora. Puoi modificarlo con qualsiasi valore compreso tra 5 minuti e 24 ore.

> **Important**
>
> Se utilizzi l'interfaccia utente ospitata e configuri un valore minimo inferiore a un'ora per i token, l'utente sarà in grado di utilizzare i token in base alla durata dei cookie di sessione, attualmente impostata su un'ora.

12. Scegli se desideri abilitare la revoca dei token per questo client dell'App. Ciò aumenterà la dimensione dei token emessi da Amazon Cognito.
13. Scegli se desideri prevenire errori di presenza degli utenti per questo client di app. Amazon Cognito risponderà alle richieste di accesso per utenti inesistenti con un messaggio generico che indica che il nome utente o la password non erano corretti.
14. Se desideri utilizzare l'interfaccia utente ospitata con questo client di app, configura Impostazioni dell'interfaccia utente ospitata.

    a. Inserisci uno o più URL di callback permessi. Questi sono gli URL web o dell'app verso cui Amazon Cognito deve reindirizzare gli utenti dopo aver completato l'autenticazione.

    b. Inserisci uno o più URL di disconnessione consentiti. Questi sono gli URL che l'app deve accettare nelle richieste a Endpoint Logout.

    c. Scegli uno o più provider di identità per consentire l'accesso degli utenti alla tua app. Puoi scegliere qualsiasi combinazione di combinazioni esistenti. IdPs Puoi autenticare gli utenti solo con il tuo pool di utenti o con una o più terze parti IdPs configurate nel tuo pool di utenti.

    d. Scegli i tipi di concessione OAuth 2.0 che devono essere accettati dal client di app.

    - Seleziona Concessione del codice di autorizzazione per trasferire i codici all'app che può riscattare in token con Endpoint Token.
    - Seleziona Concessione implicita per passare token ID e di accesso direttamente all'app. Il flusso di concessione implicita espone i token direttamente agli utenti.
    - Seleziona Credenziali del client per passare i token di accesso all'app in base alla sua conoscenza non delle credenziali utente, ma del segreto del client. Il flusso di concessione delle credenziali del cliente è reciprocamente esclusivo con il codice di autorizzazione e i flussi di concessione implicita.

    e. Seleziona gli Ambiti OpenID Connect che intendi autorizzare per l'utilizzo con questo client di app. Puoi generare token di accesso solo con l'ambito

`aws.cognito.signin.user.admin` tramite l'API dei pool di utenti. Per ambiti aggiuntivi, occorre richiedere i token di accesso dal Endpoint Token.

f. Scegli gli Ambiti personalizzati che intendi autorizzare con il client di app. Gli ambiti personalizzati vengono spesso utilizzati per autorizzare l'accesso ad API di terze parti.

15. Configura Autorizzazioni di lettura e scrittura degli attributi per questo client di app. Il client di app può disporre dell'autorizzazione per leggere e scrivere tutto, o un sottoinsieme limitato dello, schema degli attributi del pool di utenti.
16. Scegli Create app client (Crea client dell'app).
17. Fai attenzione all'ID client. Questo identificherà il client dell'App nelle richieste di registrazione e di accesso.

AWS CLI

```
aws cognito-idp create-user-pool-client --user-pool-id MyUserPoolID --client-
name myApp
```

> **Note**
>
> Utilizza il formato JSON per URL di callback e disconnessione per impedire a CLI di trattarli come file di parametri remoti:
>
> ```
> --callback-urls "["https://example.com"]"
> --logout-urls "["https://example.com"]"
> ```

Per ulteriori informazioni, consulta il riferimento ai AWS CLI comandi: create-user-pool-client

Amazon Cognito user pools API

Genera una richiesta CreateUserPoolClientAPI. È necessario specificare un valore per tutti i parametri che non si desidera impostare su un valore predefinito.

Aggiornamento del client (AWS CLI e dell' AWS API) di un'app per pool di utenti

Al AWS CLI, inserisci il seguente comando:

```
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"]"
```

Se il comando ha esito positivo, AWS CLI restituisce una conferma:

```
{
    "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
    }
}
```

Per ulteriori informazioni, vedere il riferimento ai AWS CLI comandi: update-user-pool-client.

AWS API: UpdateUserPoolClient

Ottenere informazioni su un client di app con pool di utenti (AWS CLI e AWS API)

```
aws cognito-idp describe-user-pool-client --user-pool-id MyUserPoolID --client-
id MyClientID
```

Vedi il riferimento ai AWS CLI comandi per ulteriori informazioni: describe-user-pool-client.

AWS API: DescribeUserPoolClient

Elenco di tutte le informazioni sui client dell'app in un pool di utenti (AWS CLI e AWS API)

```
aws cognito-idp list-user-pool-clients --user-pool-id "MyUserPoolID" --max-results 3
```

Vedi il riferimento ai AWS CLI comandi per ulteriori informazioni: list-user-pool-clients.

AWS API: ListUserPoolClients

Eliminazione di un client (AWS CLI e di un' AWS API) per l'app del pool di utenti

```
aws cognito-idp delete-user-pool-client --user-pool-id "MyUserPoolID" --client-id
 "MyAppClientID"
```

Per ulteriori informazioni, consulta il riferimento ai AWS CLI comandi: delete-user-pool-client

AWS API: DeleteUserPoolClient

Utilizzo dei dispositivi utente nel pool di utenti

Quando effettui l'accesso degli utenti del pool di utenti locale con l'API dei pool di utenti di Amazon Cognito, puoi associare i log delle attività degli utenti provenienti dalle funzionalità di sicurezza avanzata a ciascuno dei loro dispositivi e, facoltativamente, consentire agli utenti di ignorare l'autenticazione a più fattori (MFA) se utilizzano un dispositivo attendibile. Amazon Cognito include una chiave del dispositivo nella risposta a qualsiasi accesso che non includa già le informazioni sul dispositivo. La chiave del dispositivo è nel formato *Region*_*UUID*. Con una chiave del dispositivo, una libreria SRP (Secure Remote Password) e un pool di utenti che consente l'autenticazione del dispositivo, puoi richiedere agli utenti dell'app di definire attendibile il dispositivo corrente e non richiedere più un codice di autenticazione a più fattori (MFA) all'accesso.

Argomenti

- Configurazione dei dispositivi memorizzati
- Recupero della chiave di un dispositivo
- Accesso con un dispositivo
- Visualizzazione, aggiornamento e annullamento della memorizzazione dei dispositivi

Configurazione dei dispositivi memorizzati

Con i pool di utenti di Amazon Cognito, puoi associare ogni dispositivo degli utenti a un identificatore univoco, ovvero a una chiave del dispositivo. Specificando la chiave del dispositivo ed eseguendo l'autenticazione del dispositivo al momento dell'accesso, puoi sfruttare due funzionalità.

1. Con le funzionalità di sicurezza avanzata, puoi monitorare l'attività degli utenti su dispositivi specifici per scopi di sicurezza e analisi. Quando gli utenti accedono, l'app può autenticare ogni utente e il relativo dispositivo, aggiungendo informazioni sul dispositivo ai log delle attività.
2. La memorizzazione dei dispositivi supporta anche il flusso di autenticazione di un dispositivo attendibile, in cui gli utenti possono scegliere di accedere senza l'autenticazione a più fattori (MFA) per il periodo di tempo considerato appropriato ai requisiti di sicurezza dell'app. Quando desideri richiedere nuovamente all'utente di inviare un codice di autenticazione a più fattori (MFA), puoi modificare lo stato memorizzato del relativo dispositivo.

   I dispositivi memorizzati possono sovrascrivere l'autenticazione a più fattori (MFA) solo nei pool di utenti con l'autenticazione a più fattori (MFA) attiva.

Quando l'utente accede con un dispositivo memorizzato, è necessario eseguire un'autenticazione aggiuntiva del dispositivo durante il flusso di autenticazione. Per ulteriori informazioni, consulta Accesso con un dispositivo.

Configura la memorizzazione del dispositivo a livello di pool di utenti nella scheda Esperienza di accesso del pool di utenti, in Monitoraggio del dispositivo. Quando imposti le funzionalità dei dispositivi memorizzati nella console Amazon Cognito, hai tre opzioni: Always, User Opt-In e No.

Non memorizzare

Il pool di utenti non richiede agli utenti di memorizzare i dispositivi quando effettuano l'accesso.

Memorizza sempre

Quando l'app conferma il dispositivo di un utente, il pool di utenti ricorda sempre il dispositivo e non visualizza richieste di autenticazione a più fattori (MFA) in caso di futuri accessi riusciti da parte del dispositivo.

Accettazione dall'utente

Quando l'app conferma il dispositivo di un utente, il pool di utenti non elimina automaticamente le richieste di autenticazione a più fattori (MFA). Devi richiedere all'utente di scegliere se memorizzare o meno il proprio dispositivo.

Quando scegli Memorizza sempre o Accettazione dall'utente, Amazon Cognito genera una chiave identificativa del dispositivo e un segreto ogni volta che un utente accede da un dispositivo non identificato. La chiave del dispositivo è l'identificatore iniziale che l'app invia al pool di utenti quando l'utente esegue l'autenticazione del dispositivo.

Con ogni dispositivo utente confermato (memorizzato automaticamente o accettato dall'utente), puoi utilizzare la chiave identificativa del dispositivo e il segreto per autenticare un dispositivo a ogni accesso utente.

Puoi anche configurare le impostazioni dei dispositivi memorizzati per il pool di utenti mediante una richiesta API CreateUserPool o UpdateUserPool. Per ulteriori informazioni, consulta la sezione relativa alla proprietà DeviceConfiguration.

L'API dei pool di utenti Amazon Cognito dispone di operazioni aggiuntive per i dispositivi memorizzati.

1. ListDevices e AdminListDevices restituiscono l'elenco delle chiavi del dispositivo e i relativi metadati per un utente.
2. GetDevice e AdminGetDevice restituiscono la chiave e i metadati di un singolo dispositivo.
3. UpdateDeviceStatus e AdminUpdateDeviceStatus impostano il dispositivo di un utente come memorizzato o non memorizzato.
4. ForgetDevice e AdminForgetDevice rimuovono il dispositivo confermato di un utente dal relativo profilo.

Le operazioni API con nomi che iniziano con `Admin` sono destinate all'uso nelle app lato server e devono essere autorizzate mediante credenziali IAM. Per ulteriori informazioni, consulta Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti.

Recupero della chiave di un dispositivo

Ogni volta che l'utente accede con l'API dei pool di utenti e non include una chiave del dispositivo nei parametri di autenticazione nel formato `DEVICE_KEY`, Amazon Cognito restituisce una nuova chiave del dispositivo nella risposta. Nell'app pubblica lato client, inserisci la chiave del dispositivo nell'archivio dell'app in modo da poterla includere nelle richieste future. Nell'app riservata lato server, imposta un cookie del browser o un altro token lato client con la chiave del dispositivo dell'utente.

Prima che l'utente possa accedere con il proprio dispositivo attendibile, l'app deve confermare la chiave del dispositivo e fornire informazioni aggiuntive. Genera una richiesta ConfirmDevice ad

Amazon Cognito che confermi il dispositivo dell'utente con la relativa chiave, un nome descrittivo, un verificatore di password e un salt. Se hai configurato il pool di utenti per l'autenticazione del dispositivo tramite l'opzione Accettazione dall'utente, Amazon Cognito risponde alla richiesta `ConfirmDevice` chiedendo all'utente di scegliere se memorizzare il dispositivo corrente. Rispondi con la selezione dell'utente in una richiesta UpdateDeviceStatus.

Quando confermi il dispositivo dell'utente ma non lo imposti come dispositivo memorizzato, Amazon Cognito memorizza l'associazione ma procede con l'accesso senza dispositivo quando fornisci la chiave del dispositivo. I dispositivi possono generare log utili per la sicurezza e la risoluzione dei problemi degli utenti. Un dispositivo confermato ma non memorizzato non usa la funzionalità di accesso, ma sfrutta la funzionalità dei log di monitoraggio della sicurezza. Quando attivi le funzionalità di sicurezza avanzata per il client dell'app e codifichi il footprint del dispositivo nella tua richiesta, Amazon Cognito associa gli eventi utente al dispositivo confermato.

Recupero di una nuova chiave del dispositivo

1. Avvia la sessione di accesso dell'utente con una richiesta API InitiateAuth.
2. Rispondi a tutte le richieste di autenticazione con RespondToAuthChallenge finché non ricevi i token Web JSON (JWT) che contrassegnano la sessione di accesso dell'utente come completata.
3. Nella tua app, registra i valori restituiti da Amazon Cognito in `NewDeviceMetadata` nella relativa risposta `RespondToAuthChallenge` o `InitiateAuth`: `DeviceGroupKey` e `DeviceKey`.
4. Genera un nuovo segreto SRP per il tuo utente: un salt e un verificatore di password. Questa funzione è disponibile negli SDK che forniscono librerie SRP.
5. Richiedi all'utente il nome del dispositivo o generane uno in base alle caratteristiche del dispositivo dell'utente.
6. Fornisci il token di accesso, la chiave del dispositivo, il nome del dispositivo e il segreto SRP dell'utente in una richiesta API ConfirmDevice. Se per il pool di utenti è impostata la memorizzazione dei dispositivi mediante l'opzione Memorizza sempre, la registrazione dell'utente è completa.
7. Se Amazon Cognito ha risposto `ConfirmDevice` con `"UserConfirmationNecessary": true`, chiedi all'utente di scegliere se desidera memorizzare il dispositivo. Se l'utente sceglie di memorizzare il dispositivo, genera una richiesta API UpdateDeviceStatus con il token di accesso, la chiave del dispositivo dell'utente e `"DeviceRememberedStatus": "remembered"`.
8. Se hai richiesto ad Amazon Cognito di memorizzare il dispositivo, la prossima volta che l'utente accede, anziché una richiesta di autenticazione a più fattori (MFA), viene visualizzata una richiesta di autenticazione `DEVICE_SRP_AUTH`.

Accesso con un dispositivo

Dopo aver configurato il dispositivo di un utente in modo che venga memorizzato, Amazon Cognito non richiede più che l'utente invii un codice di autenticazione a più fattori (MFA) quando accede con la stessa chiave del dispositivo. L'autenticazione dei dispositivi si limita a sostituire la richiesta di autenticazione a più fattori (MFA) con una richiesta di autenticazione del dispositivo. Non è possibile effettuare l'accesso degli utenti solo con l'autenticazione del dispositivo. L'utente deve prima completare l'autenticazione con la propria password o una richiesta di autenticazione personalizzata. Di seguito è riportato il processo di autenticazione per un utente su un dispositivo memorizzato.

Per eseguire l'autenticazione del dispositivo in un flusso che utilizza i trigger Lambda della richiesta di autenticazione personalizzata, passa un parametro `DEVICE_KEY` nella richiesta API InitiateAuth. Dopo che l'utente ha completato con successo tutte le richieste di autorizzazione e la richiesta di autenticazione `CUSTOM_CHALLENGE` ha restituito `true` come valore `issueTokens`, Amazon Cognito restituisce un'ultima richiesta `DEVICE_SRP_AUTH`.

Accesso con un dispositivo

1. Recupera la chiave del dispositivo dell'utente dall'archivio del client.
2. Avvia la sessione di accesso dell'utente con una richiesta API InitiateAuth. Come valore di `AuthFlow` scegli `USER_SRP_AUTH`, `REFRESH_TOKEN_AUTH`, `USER_PASSWORD_AUTH` o `CUSTOM_AUTH`. In `AuthParameters`, aggiungi la chiave del dispositivo dell'utente al parametro `DEVICE_KEY` e includi gli altri parametri obbligatori per il flusso di accesso selezionato.
   a. Puoi anche passare `DEVICE_KEY` nei parametri di una risposta `PASSWORD_VERIFIER` a una richiesta di autenticazione.
3. Completa le risposte alle richieste di autenticazione finché non ricevi una richiesta `DEVICE_SRP_AUTH` nella risposta.
4. In una richiesta API RespondToAuthChallenge, invia `ChallengeName` nella richiesta `DEVICE_SRP_AUTH` e i parametri per `USERNAME`, `DEVICE_KEY` e `SRP_A`.
5. Amazon Cognito risponde con una richiesta `DEVICE_PASSWORD_VERIFIER`. La risposta a questa richiesta include i valori per `SECRET_BLOCK` e `SRP_B`.
6. Con la libreria SRP, genera e invia i parametri `PASSWORD_CLAIM_SIGNATURE`, `PASSWORD_CLAIM_SECRET_BLOCK`, `TIMESTAMP`. `USERNAME` e `DEVICE_KEY` in una richiesta `RespondToAuthChallenge` aggiuntiva.
7. Completa tutte le altre sfide finché non ricevi i token JWT dell'utente.

Il seguente pseudocodice mostra come calcolare i valori per la risposta alla richiesta `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)
```

Visualizzazione, aggiornamento e annullamento della memorizzazione dei dispositivi

Puoi implementare le seguenti funzionalità nella tua app con l'API Amazon Cognito.

1. Visualizzazione delle informazioni sul dispositivo corrente di un utente.
2. Visualizzazione dell'elenco di tutti i dispositivi dell'utente.
3. Annullamento della memorizzazione di dispositivo.
4. Aggiornamento dello stato di memorizzazione di un dispositivo.

I token di accesso che autorizzano le richieste API nelle seguenti descrizioni devono includere l'ambito `aws.cognito.signin.user.admin`. Amazon Cognito aggiunge una richiesta per questo ambito a tutti i token di accesso generati con l'API dei pool di utenti di Amazon Cognito. I gestori dell'identità digitale di terze parti devono gestire separatamente i dispositivi e l'autenticazione a più fattori (MFA) dei propri utenti che effettuano l'autenticazione su Amazon Cognito. Nell'interfaccia utente ospitata, puoi richiedere l'ambito `aws.cognito.signin.user.admin`, ma l'interfaccia utente ospitata aggiunge automaticamente le informazioni sul dispositivo ai log utente di sicurezza avanzata e non consente la memorizzazione dei dispositivi.

Visualizzazione delle informazioni su un dispositivo

È possibile eseguire una query sulle informazioni sul dispositivo di un utente per determinare se è ancora in uso. Ad esempio, potresti voler disattivare i dispositivi memorizzati se non hanno effettuato l'accesso per 90 giorni consecutivi.

- Per visualizzare le informazioni sul dispositivo dell'utente in un'app client pubblica, invia la chiave di accesso e la chiave del dispositivo dell'utente in una richiesta API GetDevice.

- Per visualizzare le informazioni sul dispositivo dell'utente in un'app client riservata, firma una richiesta API AdminGetDevice con le credenziali AWS e invia il nome utente, la chiave del dispositivo e il pool di utenti dell'utente.

Visualizzazione dell'elenco di tutti i dispositivi dell'utente.

Puoi visualizzare l'elenco di tutti i dispositivi dell'utente e delle relative proprietà. Ad esempio, potresti voler verificare se il dispositivo corrente corrisponde a un dispositivo memorizzato.

- In un'app client pubblica, invia il token di accesso dell'utente in una richiesta API ListDevices.
- In un'app client riservata, firma una richiesta API AdminListDevices con le credenziali AWS e invia il nome utente e il pool di utenti dell'utente.

Annullamento della memorizzazione di un dispositivo

Puoi eliminare la chiave del dispositivo di un utente. È consigliabile eseguire questa operazione quando un utente non utilizza più un dispositivo o quando vengono rilevate attività insolite e si desidera richiedere a un utente di rieseguire l'autenticazione a più fattori (MFA). Per registrare nuovamente il dispositivo in un secondo momento, è necessario generare e archiviare una nuova chiave del dispositivo.

- In un'app client pubblica, invia la chiave del dispositivo e il token di accesso dell'utente nella richiesta API ForgetDevice.
- In un'app client riservata, invia la chiave del dispositivo e il token di accesso dell'utente nella richiesta API AdminForgetDevice.

Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti

Quando desideri eseguire la registrazione, l'accesso e la gestione di utenti nel pool di utenti, puoi utilizzare due opzioni.

1. Gli endpoint del pool di utenti includono l'interfaccia utente ospitata e gli endpoint di federazione. Costituiscono un pacchetto di pagine web pubbliche che vengono attivate da Amazon Cognito quando scegli un dominio per il pool di utenti. Per una guida rapida alle funzionalità di autenticazione e autorizzazione dei pool di utenti di Amazon Cognito, incluse le pagine per la

registrazione, l'accesso, la gestione delle password e l'autenticazione a più fattori (MFA), utilizza l'interfaccia utente integrata dell'interfaccia utente ospitata. Gli altri endpoint del pool di utenti facilitano l'autenticazione con gestori dell'identità digitale di terze parti. I servizi che vengono eseguiti includono quanto segue.

   a. Endpoint di callback dei provider di servizi per richieste autenticate dagli IdP, come `saml2/idpresponse` e `oauth2/idpresponse`. Quando Amazon Cognito è un provider di servizi (SP) intermedio tra l'app e l'IdP, gli endpoint di callback rappresentano il servizio.

   b. Endpoint che forniscono informazioni sull'ambiente, ad esempio `oauth2/userInfo` e `jwks.json`. L'app utilizza questi endpoint quando verifica i token o recupera i dati del profilo utente con SDK AWS e librerie OAuth 2.0.

2. L'API dei pool di utenti Amazon Cognito è un insieme di strumenti per l'app web o per dispositivi mobili, dopo che raccoglie le informazioni di accesso nel front-end personalizzato, per autenticare gli utenti. L'autenticazione API dei pool di utenti produce i seguenti token web JSON.

   a. Un token di identità con richieste di attributi verificabili da parte dell'utente.

   b. Un token di accesso che autorizza l'utente a creare richieste API autorizzate da token a un endpoint di servizio AWS.

      > **Note**
      >
      > Per impostazione predefinita, i token di accesso dell'autenticazione API dei pool di utenti contengono solo l'ambito `aws.cognito.signin.user.admin`. Se desideri generare un token di accesso con altri ambiti, ad esempio per autorizzare una richiesta a un'API di terze parti, richiedi gli ambiti durante l'autenticazione tramite gli endpoint del pool di utenti o aggiungi ambiti personalizzati in un Trigger Lambda di pre-generazione del token. La personalizzazione dei token di accesso aggiunge costi alla fattura AWS.

Puoi collegare un utente federato, che normalmente accede tramite gli endpoint dei pool di utenti, con un utente il cui profilo è locale al pool di utenti. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno. Se colleghi l'identità federata a un utente locale in una richiesta API AdminLinkProviderForUser, l'utente può accedere con l'API dei pool di utenti. Per ulteriori informazioni, consulta Collegamento di utenti federati a un profilo utente esistente.

L'API dei pool di utenti di Amazon Cognito ha un doppio scopo. Consente di creare e configurare le risorse dei pool di utenti di Amazon Cognito. Ad esempio, è possibile creare pool di utenti, aggiungere trigger AWS Lambda e configurare il dominio dell'interfaccia utente ospitata. Inoltre, l'API dei pool

di utenti esegue operazioni di registrazione, accesso e altre operazioni utente per utenti locali e collegati.

Scenario di esempio con l'API dei pool di utenti di Amazon Cognito

1. L'utente seleziona il pulsante "Crea un account" creato nell'app. Inserisce un indirizzo e-mail e una password.
2. L'app invia una richiesta API SignUp e crea un nuovo utente nel pool di utenti.
3. L'app richiede all'utente un codice di conferma e-mail. Gli utenti inseriscono il codice ricevuto in un messaggio di posta elettronica.
4. L'app invia una richiesta API ConfirmSignUp con il codice di conferma dell'utente.
5. L'app richiede all'utente il nome utente e password. Le relative informazioni vengono inserite.
6. L'app invia una richiesta API InitiateAuth e memorizza un token ID, un token di accesso e un token di aggiornamento. L'app chiama le librerie OIDC per gestire i token di un utente e mantenere persistente la sessione dell'utente.

Nell'API dei pool di utenti di Amazon Cognito, non è possibile consentire l'accesso agli utenti che eseguono la federazione tramite un IdP. Questi utenti devono essere autenticati tramite gli endpoint del pool di utenti. Per ulteriori informazioni sugli endpoint del pool di utenti che includono l'interfaccia utente ospitata, consultare Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata. Gli utenti federati possono accedere all'interfaccia utente ospitata e selezionare il provider di identità oppure puoi inviare gli utenti direttamente al provider di identità per l'accesso, ignorando l'interfaccia utente ospitata. Quando la richiesta API a Endpoint Authorize include un parametro IdP, Amazon Cognito reindirizza automaticamente l'utente alla pagina di accesso IdP.

Scenario di esempio con endpoint del pool di utenti

1. L'utente seleziona il pulsante "Crea un account" creato nell'app.
2. L'utente riceve un elenco di provider di identità social in cui sono state registrate le credenziali sviluppatore. L'utente sceglie Apple.
3. L'app avvia una richiesta a Endpoint Authorize con il nome di provider `SignInWithApple`.
4. Il browser dell'utente apre la pagina di autorizzazione Apple OAuth. L'utente sceglie di consentire ad Amazon Cognito di leggere le informazioni sul profilo.
5. Amazon Cognito conferma il token di accesso Apple ed esegue una query sul profilo Apple dell'utente.

6. L'utente presenta un codice di autorizzazione Amazon Cognito all'app.
7. L'app scambia il codice di autorizzazione con Endpoint Token e memorizza un token ID, un token di accesso e un token di aggiornamento. L'app chiama le librerie OIDC per gestire i token di un utente e mantenere persistente la sessione dell'utente.

L'API dei pool di utenti e gli endpoint del pool di utenti supportano un'ampia gamma di scenari, descritti in questa guida. Nelle sezioni seguenti viene illustrato in che modo l'API dei pool di utenti si divide ulteriormente in classi che supportano i requisiti di registrazione, accesso e gestione delle risorse.

Operazioni API autenticate e non autenticate per pool di utenti di Amazon Cognito

L'API dei pool di utenti di Amazon Cognito, un'interfaccia di gestione delle risorse e un'interfaccia di autenticazione e autorizzazione lato utente, combina i modelli di autorizzazione che seguono le relative operazioni. A seconda dell'operazione API, potrebbe essere necessario fornire l'autorizzazione con credenziali IAM, un token di accesso, un token di sessione, un segreto del client o una combinazione di questi. Per molte operazioni di autenticazione e autorizzazione utente, è possibile scegliere tra versioni autenticate e non autenticate della richiesta. Le operazioni non autenticate sono best practice di sicurezza per le app distribuite agli utenti, come app per dispositivi mobili; non è necessario includere alcun segreto nel codice.

È possibile assegnare autorizzazioni solo nelle policy IAM per Operazioni di gestione autenticate IAM e Operazioni utente autenticate IAM.

Operazioni di gestione autenticate IAM

Le operazioni di gestione autenticate IAM modificano e visualizzano il pool di utenti e la configurazione del client dell'app, come si farebbe in AWS Management Console.

Ad esempio, per modificare il pool di utenti in una richiesta API UpdateUserPool, è necessario presentare le credenziali AWS e le autorizzazioni IAM per aggiornare la risorsa.

Per autorizzare queste richieste nell'AWS Command Line Interface (AWS CLI) o in un SDK AWS, configura il tuo ambiente con variabili di ambiente o configurazione client che aggiunge credenziali IAM alla tua richiesta. Per ulteriori informazioni, consulta Accesso a AWS tramite le proprie credenziali AWS nella Riferimenti generali di AWS. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. È necessario autorizzare o firmare

queste richieste con le credenziali AWS integrate nell'intestazione della richiesta. Per informazioni, consulta Firma delle richieste API AWS.

Operazioni di gestione autenticate IAM

`AddCustomAttributes`

`CreateGroup`

`CreateIdentityProvider`

`CreateResourceServer`

`CreateUserImportJob`

`CreateUserPool`

`CreateUserPoolClient`

`CreateUserPoolDomain`

`DeleteGroup`

`DeleteIdentityProvider`

`DeleteResourceServer`

`DeleteUserPool`

`DeleteUserPoolClient`

`DeleteUserPoolDomain`

`DescribeIdentityProvider`

`DescribeResourceServer`

`DescribeRiskConfiguration`

`DescribeUserImportJob`

`DescribeUserPool`

Operazioni di gestione autenticate IAM

`DescribeUserPoolClient`

`DescribeUserPoolDomain`

`GetCSVHeader`

`GetGroup`

`GetIdentityProviderByIdentifier`

`GetSigningCertificate`

`GetUICustomization`

`GetUserPoolMfaConfig`

`ListGroups`

`ListIdentityProviders`

`ListResourceServers`

`ListTagsForResource`

`ListUserImportJobs`

`ListUserPoolClients`

`ListUserPools`

`ListUsers`

`ListUsersInGroup`

`SetRiskConfiguration`

`SetUICustomization`

`SetUserPoolMfaConfig`

`StartUserImportJob`

Operazioni di gestione autenticate IAM

`StopUserImportJob`

`TagResource`

`UntagResource`

`UpdateGroup`

`UpdateIdentityProvider`

`UpdateResourceServer`

`UpdateUserPool`

`UpdateUserPoolClient`

`UpdateUserPoolDomain`

Operazioni utente autenticate IAM

Operazioni utente autenticate IAM: registrazione, accesso, gestione delle credenziali, modifica e visualizzazione degli utenti.

Ad esempio, un livello applicazione lato server potrebbe eseguire il backup di un front-end Web. L'app lato server è un client riservato OAuth considerato affidabile con accesso privilegiato alle risorse Amazon Cognito. Per registrare un utente nell'app, il server può includere credenziali AWS in una richiesta API AdminCreateUser. Per ulteriori informazioni sui tipi di client OAuth, consulta la sezione relativa ai tipi di client in The OAuth 2.0 Authorization Framework.

Per autorizzare queste richieste nella AWS CLI o in un SDK AWS, configura il tuo ambiente app lato server con variabili di ambiente o una configurazione client che aggiunge credenziali IAM alla tua richiesta. Per ulteriori informazioni, consulta Accesso a AWS tramite le proprie credenziali AWS nella Riferimenti generali di AWS. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. È necessario autorizzare o firmare queste richieste con le credenziali AWS integrate nell'intestazione della richiesta. Per informazioni, consulta Firma delle richieste API AWS.

Se il client dell'app dispone di un segreto del client, devi fornire le tue credenziali IAM e, a seconda dell'operazione, il parametro `SecretHash` o il valore `SECRET_HASH` in `AuthParameters`. Per ulteriori informazioni, consulta Calcolo dei valori SecretHash.

Operazioni utente autenticate IAM
`AdminAddUserToGroup`
`AdminConfirmSignUp`
`AdminCreateUser`
`AdminDeleteUser`
`AdminDeleteUserAttributes`
`AdminDisableProviderForUser`
`AdminDisableUser`
`AdminEnableUser`
`AdminForgetDevice`
`AdminGetDevice`
`AdminGetUser`
`AdminInitiateAuth`
`AdminLinkProviderForUser`
`AdminListDevices`
`AdminListGroupsForUser`
`AdminListUserAuthEvents`
`AdminRemoveUserFromGroup`
`AdminResetUserPassword`

Operazioni utente autenticate IAM

`AdminRespondToAuthChallenge`

`AdminSetUserMFAPreference`

`AdminSetUserPassword`

`AdminSetUserSettings`

`AdminUpdateAuthEventFeedback`

`AdminUpdateDeviceStatus`

`AdminUpdateUserAttributes`

`AdminUserGlobalSignOut`

Operazioni utente non autenticate

Operazioni utente non autenticate: registrazione, accesso e avvio del ripristino delle password per gli utenti. Utilizza operazioni API non autenticate, o pubbliche, per consentire a chiunque su Internet di eseguire la registrazione e l'accesso all'app.

Ad esempio, per registrare un utente nell'app, puoi distribuire un client pubblico OAuth che non fornisce alcun accesso privilegiato ai segreti. Puoi registrare questo utente con l'operazione API non autenticata SignUp.

Per inviare queste richieste in un client pubblico sviluppato con un SDK AWS, non è necessario configurare alcuna credenziale. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito senza autorizzazione aggiuntiva.

Se il client dell'app dispone di un segreto del client, devi fornire, a seconda dell'operazione, il parametro `SecretHash` o il valore `SECRET_HASH` in `AuthParameters`. Per ulteriori informazioni, consulta Calcolo dei valori SecretHash.

Operazioni utente non autenticate

`SignUp`

Operazioni utente non autenticate

`ConfirmSignUp`

`ResendConfirmationCode`

`ForgotPassword`

`ConfirmForgotPassword`

`InitiateAuth`

Operazioni utente autorizzate tramite token

Le operazioni utente autorizzate tramite token consentono di disconnettersi, gestire le credenziali, modificare e visualizzare gli utenti dopo che hanno effettuato l'accesso o iniziato il processo di accesso. Utilizza le operazioni API autorizzate tramite token quando non desideri distribuire segreti nella tua app e desideri autorizzare le richieste con le credenziali dell'utente. Se l'utente ha completato l'accesso, devi autorizzare la richiesta API autorizzata tramite token con un token di accesso. Se l'utente sta eseguendo un processo di accesso, devi autorizzare la relativa richiesta API autorizzata tramite token con un token di sessione restituito da Amazon Cognito nella risposta alla richiesta precedente.

Ad esempio, in un client pubblico, potrebbe essere necessario aggiornare il profilo di un utente in modo da limitare l'accesso in scrittura solo al profilo dell'utente. Per effettuare questo aggiornamento, il client può includere il token di accesso dell'utente in una richiesta API UpdateUserAttributes.

Per inviare queste richieste in un client pubblico sviluppato con un SDK AWS, non è necessario configurare alcuna credenziale. Includi un parametro `AccessToken` o `Session` nella richiesta. Puoi anche inviare richieste direttamente agli endpoint del servizio per l'API dei pool di utenti Amazon Cognito. Per autorizzare una richiesta a un endpoint del servizio, includi il token di accesso o sessione nel corpo POST della richiesta.

Per firmare una richiesta API per un'operazione autorizzata tramite token, includi il token di accesso come un'intestazione `Authorization` nella richiesta, nel formato `Bearer` *`<Base64-encoded access token>`*.

Operazioni utente autorizzate tramite token	AccessToken	Sessione
`RespondToAuthChallenge`		✓
`ChangePassword`	✓	
`GetUser`	✓	
`UpdateUserAttributes`	✓	
`DeleteUserAttributes`	✓	
`DeleteUser`	✓	
`ConfirmDevice`	✓	
`ForgetDevice`	✓	
`GetDevice`	✓	
`ListDevices`	✓	
`UpdateDeviceStatus`	✓	
`GetUserAttributeVerificationCode`	✓	
`VerifyUserAttribute`	✓	
`SetUserSettings`	✓	

Operazioni utente autorizzate tramite token	AccessToken	Sessione
`SetUserMFAPreference`	✓	
`GlobalSignOut`	✓	
`AssociateSoftwareToken`	✓	✓
`UpdateAuthEventFeedback`		✓
`VerifySoftwareToken`	✓	✓
`RevokeToken` [1]		

[1] RevokeToken accetta un token di aggiornamento come un parametro. Il token di aggiornamento funge da token di autorizzazione e come risorsa di destinazione.

Aggiornamento della configurazione del pool di utenti

Per modificare le impostazioni dei pool di utenti di Amazon Cognito in AWS Management Console, naviga tra le schede basate sulle funzionalità nelle impostazioni del pool di utenti e aggiorna i campi come descritto in altre aree di questa guida. Dopo aver creato un pool di utenti, non puoi più modificare alcune impostazioni. Se desideri modificare le seguenti impostazioni, devi creare un nuovo pool di utenti o un nuovo client dell'app.

Nome del bacino d’utenza

Nome del parametro API: PoolName

Nome descrittivo assegnato al pool di utenti. Per modificare il nome di un pool di utenti, devi crearne uno nuovo.

Opzioni di accesso al pool di utenti di Amazon Cognito

Nomi dei parametri API: AliasAttributese UsernameAttributes

Gli attributi che gli utenti possono passare come nome utente quando effettuano l'accesso. Quando crei un pool di utenti, puoi scegliere di consentire l'accesso utilizzando il nome utente, l'indirizzo e-mail, il numero di telefono o il nome utente preferito. Per modificare le opzioni di accesso del pool di utenti, devi crearne uno nuovo.

Make user name case sensitive (Il nome utente rispetta la distinzione tra maiuscole e minuscole)

Nome del parametro API: UsernameConfiguration

Quando crei un nome utente corrispondente a un altro nome utente ma con una diversa combinazione di maiuscole e minuscole, Amazon Cognito può considerare tali nomi come lo stesso utente o come utenti univoci. Per ulteriori informazioni, consulta Distinzione tra maiuscole e minuscole del bacino d'utenza. Per modificare la distinzione tra lettere maiuscole e minuscole, devi creare uno nuovo pool di utenti.

Client secret

Nome del parametro API: GenerateSecret

Quando crei un client dell'app, puoi generare un segreto del client in modo che solo le origini attendibili possano inviare richieste al pool di utenti. Per ulteriori informazioni, consulta Client dell'app pool di utenti. Per modificare il segreto di un client, crea un nuovo client dell'app nello stesso pool di utenti.

Attributi obbligatori

Nome del parametro API: Schema

Attributi per i quali gli utenti devono fornire valori quando si registrano o quando tali attributi vengono creati. Per ulteriori informazioni, consulta Attributi del bacino d'utenza. Per modificare gli attributi richiesti, devi creare un nuovo pool di utenti.

Attributi personalizzati

Nome del parametro API: Schema

Attributi con nomi personalizzati. È possibile modificare il valore dell'attributo personalizzato di un utente, ma non è possibile eliminare un attributo personalizzato dal pool di utenti. Per ulteriori

informazioni, consulta Attributi del bacino d'utenza. Se si raggiunge il numero massimo di attributi personalizzati e si desidera modificare l'elenco, creare un nuovo pool di utenti.

Configurazione SMS

Dopo aver attivato i messaggi SMS nel tuo pool di utenti, non puoi disattivarli.

- Se scegli di configurare i messaggi SMS quando crei un pool di utenti, non puoi disattivare gli SMS dopo aver completato la configurazione.
- Puoi attivare i messaggi SMS in un pool di utenti che hai creato, ma dopo non puoi disattivare gli SMS.
- Amazon Cognito può utilizzare i messaggi SMS per l'invito e il ripristino degli account utente, la verifica degli attributi e l'autenticazione a più fattori (MFA). Dopo aver attivato i messaggi SMS, puoi attivare o disattivare i messaggi SMS per queste funzioni in qualsiasi momento.
- La configurazione dei messaggi SMS include un ruolo IAM che deleghi ad Amazon Cognito per inviare messaggi con Amazon SNS. Puoi modificare il ruolo assegnato in qualsiasi momento.

Aggiornamento di un pool di utenti con un AWS SDK o un' AWS CDK API REST

Nella console Amazon Cognito, puoi modificare le impostazioni del pool di utenti un parametro alla volta. Ad esempio, per aggiungere un trigger Lambda, scegli Aggiungi trigger Lambda e scegli la funzione e il tipo di trigger. L'API dei pool di utenti di Amazon Cognito è strutturata in modo tale che le operazioni di aggiornamento per i pool di utenti e i client delle app richiedano il set completo di parametri per il pool di utenti. Tuttavia, la console automatizza in modo trasparente questa operazione di aggiornamento con le altre impostazioni del pool di utenti.

A volte è possibile che una modifica apportata in un altro punto dell'utente Account AWS possa causare la generazione di un errore negli aggiornamenti quando non sono correlati all'impostazione che si desidera modificare. Un'identità Amazon SES eliminata o una modifica di un'autorizzazione IAM per AWS WAF, ad esempio. Se uno dei parametri correnti non è più valido, non puoi aggiornare le impostazioni finché non lo correggi. Quando riscontri un errore di questo tipo, esamina la risposta all'errore e convalida l'impostazione menzionata.

I pool di utenti di Amazon Cognito AWS Cloud Development Kit (AWS CDK), l'API REST e gli AWS SDK sono strumenti per l'automazione e la configurazione programmatica delle risorse di

Amazon Cognito. Le richieste con questi strumenti devono inoltre, come la console Amazon Cognito, aggiornare un'impostazione con una configurazione completa delle risorse nel corpo della richiesta. A un livello elevato, è necessario eseguire la seguente procedura.

1. Acquisisci l'output di un'operazione che descrive la configurazione della risorsa esistente.
2. Modifica l'output modificando le impostazioni.
3. Invia la configurazione modificata con un'operazione che aggiorni la tua risorsa.

La procedura seguente aggiorna la configurazione con l'operazione UpdateUserPoolAPI. Lo stesso approccio, con campi di input diversi, si applica a UpdateUserPoolClient.

> **Important**
>
> Se non fornisci valori per i parametri esistenti, Amazon Cognito li imposta sui valori predefiniti. Ad esempio, se è già presente la proprietà `LambdaConfig` e si invia un'operazione `UpdateUserPool` con una proprietà `LambdaConfig` vuota, elimina l'assegnazione di tutte le funzioni Lambda ai trigger del pool di utenti. Pianifica di conseguenza quando vuoi automatizzare le modifiche alla configurazione del pool di utenti.

1. Acquisisci lo stato esistente del tuo pool di utenti con DescribeUserPool.
2. Formatta l'output di `DescribeUserPool` per abbinare i parametri della richiesta di`UpdateUserPool`. Rimuovi i seguenti campi di livello superiore e i relativi oggetti figlio dal JSON di output.

   - `Arn`
   - `CreationDate`
   - `CustomDomain`
     - Aggiorna questo campo con l'operazione UpdateUserPoolDomainAPI.
   - `Domain`
     - Aggiorna questo campo con l'operazione UpdateUserPoolDomainAPI.
   - `EmailConfigurationFailure`
   - `EstimatedNumberOfUsers`
   - `Id`
   - `LastModifiedDate`

- `Name`
- `SchemaAttributes`
- `SmsConfigurationFailure`
- `Status`

3. Conferma che il JSON risultante corrisponda ai parametri della richiesta di `UpdateUserPool`.
4. Modifica tutti i parametri che desideri modificare nel JSON risultante.
5. Invia una richiesta API `UpdateUserPool` con il JSON modificato come input della richiesta.

È possibile utilizzare anche questo output `DescribeUserPool` modificato nel parametro `--cli-input-json` di `update-user-pool` nella AWS CLI.

In alternativa, esegui il AWS CLI comando seguente per generare JSON con valori vuoti per i campi di input accettati per. `update-user-pool` È quindi possibile popolare questi campi con i valori esistenti del pool di utenti.

```
aws cognito-idp update-user-pool --generate-cli-skeleton --output json
```

Esegui il comando riportato di seguito per generare lo stesso oggetto JSON per un client dell'app.

```
aws cognito-idp update-user-pool-client --generate-cli-skeleton --output json
```

Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione

Un pool di utenti Amazon Cognito con un dominio è un server di autorizzazione conforme a OAuth-2.0 e un' ready-to-use interfaccia utente ospitata (UI) per l'autenticazione. Il server di autorizzazione instrada le richieste di autenticazione, emette e gestisce i token Web JSON (JWT) e fornisce informazioni sugli attributi utente. L'interfaccia utente ospitata è una raccolta di interfacce Web per attività di base di registrazione, accesso, autenticazione a più fattori e reimpostazione della password nel pool di utenti. È anche un hub centrale per l'autenticazione con i provider di identità di terze parti () IdPs che associ alla tua app. L'app può richiamare l'interfaccia utente ospitata e gli endpoint di autorizzazione quando desideri autenticare e autorizzare gli utenti. Puoi adattare l'esperienza utente dell'interfaccia utente ospitata al tuo marchio con il tuo logo e la personalizzazione CSS. Per ulteriori informazioni sui componenti dell'interfaccia utente ospitata e del server di

autorizzazione, consulta Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata.

> **Note**
>
> L'interfaccia utente ospitata di Amazon Cognito non supporta l'autenticazione personalizzata con trigger Lambda di richiesta di autenticazione personalizzati.

Argomenti

- Configurazione dell'interfaccia utente ospitata con AWS Amplify
- Configurazione dell'interfaccia utente ospitata con la console Amazon Cognito
- Visualizzazione della pagina di accesso
- Dettagli sull'interfaccia utente ospitata dei pool di utenti di Amazon Cognito
- Configurazione di un dominio di bacino d'utenza
- Personalizzazione delle pagine Web di registrazione e accesso integrate
- Registrazione e accesso con l'interfaccia utente ospitata

Configurazione dell'interfaccia utente ospitata con AWS Amplify

Se usi AWS Amplify per aggiungere l'autenticazione alla tua app web o mobile, puoi configurare l'interfaccia utente ospitata utilizzando l'interfaccia a riga di comando (CLI) e le librerie nel AWS Amplify framework. Per aggiungere l'autenticazione all'App, utilizzare la CLI AWS Amplify per aggiungere la categoria `Auth` al progetto. Quindi, nel codice client, usi le AWS Amplify librerie per autenticare gli utenti con il tuo pool di utenti Amazon Cognito.

È possibile visualizzare un'interfaccia utente ospitata preconfigurata oppure è possibile federare gli utenti tramite un endpoint OAuth 2.0 che reindirizza a un provider di accesso social, ad esempio Facebook, Google, Amazon o Apple. Dopo che un utente ha eseguito correttamente l'autenticazione con il provider di contenuti social, AWS Amplify crea un nuovo utente nel bacino d'utenza, se necessario, e fornisce all'app il token OIDC dell'utente.

Gli esempi seguenti mostrano come AWS Amplify configurare l'interfaccia utente ospitata con i provider di social network nell'app.

- AWS Amplify autenticazione per JavaScript.
- AWS Amplify autenticazione per Swift.

- AWS Amplify autenticazione per Flutter.
- AWS Amplify autenticazione per Android.

Configurazione dell'interfaccia utente ospitata con la console Amazon Cognito

Creazione di un client dell'App

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Seleziona la scheda App integration (Integrazione App).
5. Alla voce App clients (Client dell'App), seleziona Create an App client (Crea un client dell'App).
6. Selezionare un App type (Tipo di App): Public client (Client pubblico), Confidential client (Client riservato), oppure Other (Altro). Un client pubblico in genere opera dai dispositivi degli utenti e utilizza API non autenticate e autenticate con token. Un client riservato in genere opera da un'app su un server centrale di cui ti fidi con segreti client e credenziali API e utilizza intestazioni e credenziali di autorizzazione per firmare le richieste. AWS Identity and Access Management Se il tuo caso d'uso è diverso dalle impostazioni preconfigurate del client dell'App per un Public client (Client pubblico) o Confidential client (Client riservato), seleziona Other (Altro).
7. Inserisci un nome del client dell'App.
8. Seleziona i Authentication flows (Flussi di autenticazione) che desideri per entrare nel tuo client App.
9. Configurazione di Authentication flow session duration (Durata della sessione del flusso di autenticazione). Questo è il tempo a disposizione degli utenti per completare ogni verifica di autenticazione prima della scadenza del token di sessione.
10. (Facoltativo) Configura la scadenza del token.

    a. Specifica la scadenza del token di aggiornamento del client dell'App. Il valore predefinito è 30 giorni. Puoi modificarlo con qualsiasi valore compreso tra 1 ora e 10 anni.

    b. Specifica la scadenza del token di accesso del client dell'App. Il valore predefinito è 1 ora. Puoi modificarlo con qualsiasi valore compreso tra 5 minuti e 24 ore.

    c. Specifica la scadenza del token ID del client dell'App. Il valore predefinito è 1 ora. Puoi modificarlo con qualsiasi valore compreso tra 5 minuti e 24 ore.

> **Important**
>
> Se utilizzi l'interfaccia utente ospitata e configuri un valore minimo inferiore a un'ora per i token, l'utente sarà in grado di utilizzare i token in base alla durata dei cookie di sessione, attualmente impostata su un'ora.

11. Scegli Generate client secret (Genera segreto client) per fare in modo che Amazon Cognito generi un segreto client per te. I segreti client sono generalmente associati ai client riservati.
12. Scegli se vuoi abilitare l'opzione Enable token revocation (Abilita la revoca dei token) per questo client dell'App. Ciò aumenterà le dimensioni dei token. Per ulteriori informazioni, consulta Revoking Tokens (Revoca dei token).
13. Scegli se desideri impedire messaggi di errore che rivelano l'esistenza dell'utente per questo client dell'App. Amazon Cognito risponderà alle richieste di accesso per utenti inesistenti con un messaggio generico che indica che il nome utente o la password non erano corretti.
14. (Facoltativo) Configura le autorizzazioni di lettura e scrittura dell'attributo) per questo client dell'App. Il client dell'App può avere l'autorizzazione di leggere e scrivere solo un sottoinsieme limitato dello schema degli attributi del bacino d'utenza.
15. Scegli Create (Crea).
16. Fai attenzione all'ID client. Questo identificherà il client dell'App nelle richieste di registrazione e di accesso.

Configurazione dell'app

1. Nella scheda App integration (Integrazione app), seleziona il client dell'App alla voce App clients (Client dell'app). Controlla le informazioni attuali sull'Hosted UI (Interfaccia utente ospitata).
2. Add a callback URL (Aggiunta di un URL di callback) alla voce Allowed callback URL(s) (Url di callback consentiti). Un URL di callback indica dove l'utente deve essere reindirizzato dopo aver effettuato correttamente l'accesso.
3. Add a sign-out URL (Aggiungi un URL di disconnessione) alla voce Allowed sign-out URL(s) (Url di disconnessione consentiti). Un URL di disconnessione indica dove l'utente deve essere reindirizzato dopo la disconnessione.
4. Aggiungere almeno una delle voci dell'elenco di opzioni elencate in Identity providers (Provider di identità).

5. Alla voce OAuth 2.0 grant types (Tipi di concessione OAuth 2.0) seleziona Authorization code grant (Concessione del codice di autorizzazione) per restituire un codice di autorizzazione che viene quindi scambiato per i token dei bacini d'utenza. Poiché i token non vengono mai esposti direttamente a un utente finale, vi sono meno probabilità che vengano compromessi. Tuttavia, per un'applicazione personalizzata è obbligatorio nel back-end scambiare il codice di autorizzazione per i token dei bacini d'utenza. Per motivi di sicurezza, ti suggeriamo di utilizzare il flusso di concessione del codice di autorizzazione insieme al protocollo Proof Key for Code Exchange (PKCE) per le app mobili.
6. Alla voce OAuth 2.0 grant types (Tipi di concessione OAuth 2.0), seleziona Implicit grant (Concessione implicita) per fare in modo che Amazon Cognito restituisca i token Web JSON (JWT) del bacino d'utenza. È possibile usare questo flusso quando non è disponibile un back-end per scambiare un codice di autorizzazione per i token. È inoltre utile per il debug dei token.
7. È possibile abilitare sia la Authorization code grant (Concessione del codice di autorizzazione) sia la Implicit code grant (Concessione implicita del codice) e usare entrambe in base alle esigenze. Se né il le concessioni per il Codice di autorizzazione né quelle per il Codice implicito sono selezionate e il client dell'app ha un segreto client, puoi abilitare le concessioni delle Client credentials (Credenziali del client). Seleziona le Client credentials (Credenziali del client) solo se è necessario che l'app richieda i token di accesso a proprio nome e non a nome di un utente.
8. Seleziona gli OpenID Connect scopes (ambiti di OpenID Connect) che intendi autorizzare per questo client dell'app.
9. Scegli Save changes (Salva modifiche).

Configura un dominio

1. Passa alla scheda App Integration (Integrazione App) per il bacino d'utenza.
2. Accanto a Dominio, seleziona Operazioni, quindi seleziona Create custom domain (Crea dominio personalizzato) o Create Cognito domain (Crea dominio Cognito). Se hai già configurato un dominio del bacino d'utenza, scegli l'opzione Delete Cognito domain (Elimina dominio Cognito) o Delete custom domain (Elimina dominio personalizzato) prima di creare un nuovo dominio personalizzato.
3. Inserisci un prefisso dominio disponibile da utilizzare con un Dominio Cognito. Per informazioni sulla configurazione di un Dominio personalizzato, consulta la sezione Utilizzo del proprio dominio per l'interfaccia utente ospitata
4. Scegli Create (Crea).

Visualizzazione della pagina di accesso

Nella console Amazon Cognito, seleziona il pulsante View Hosted UI (Visualizza l'interfaccia utente ospitata), nell'area App clients and analytics (Client di app e analisi dei dati) nella scheda App integration (Integrazione app). Questo pulsante ti reindirizza alla pagina di accesso nell'interfaccia utente ospitata con i seguenti parametri di base.

- ID del client dell'app
- Richiesta di concessione del codice di autorizzazione
- Richiesta per tutti gli ambiti attivati per il client dell'app corrente
- Primo URL di callback nell'elenco per il client dell'app corrente

Il pulsante View hosted UI (Visualizza l'interfaccia utente ospitata) è utile quando si desidera testare le funzioni di base dell'interfaccia utente ospitata. Puoi personalizzare l'URL di accesso con parametri aggiuntivi e modificati. Nella maggior parte dei casi, i parametri generati automaticamente dal link View hosted UI (Visualizza l'interfaccia utente ospitata) non soddisfano completamente le esigenze della app. In questi casi, devi personalizzare l'URL richiamato dall'app in fase di accesso degli utenti. Per ulteriori informazioni sui parametri di accesso e sui relativi valori, consulta Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata.

La pagina Web di accesso dell'interfaccia utente ospitata utilizza il seguente formato URL. In questo esempio viene richiesta una concessione del codice di autorizzazione con il parametro `response_type=code`.

```
https://<your domain>/oauth2/authorize?response_type=code&client_id=<your app client
 id>&redirect_uri=<your callback url>
```

Puoi recuperare la stringa di dominio del pool di utenti dalla scheda Integrazione di app. Nella stessa scheda, puoi identificare gli ID client dell'app, i relativi URL di callback, gli ambiti consentiti e altre configurazioni in Client di app e analisi dei dati.

Quando si accede all'endpoint `/oauth2/authorize` con i parametri personalizzati, Amazon Cognito ti reindirizza all'endpoint `/oauth2/login` o, se è presente un parametro `identity_provider` o `idp_identifier`, ti reindirizza in modalità invisibile all'utente alla pagina di accesso del gestore dell'identità digitale (IdP). Per un URL di esempio che ignora l'interfaccia utente ospitata, consulta Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito.

Esempio di richiesta di interfaccia utente ospitata per una concessione implicita

Puoi visualizzare la pagina Web di accesso dell'interfaccia utente ospitata tramite il seguente URL per la concessione implicita del codice dove `response_type=token`. Dopo aver eseguito correttamente l'accesso, Amazon Cognito restituisce i token del bacino d'utenza alla barra degli indirizzi del browser Web.

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

I token di identità e accesso vengono visualizzati come parametri aggiunti all'URL di reindirizzamento.

Di seguito è riportato un esempio di risposta da una richiesta di concessione implicita.

```
            https://mydomain.example.com/
#id_token=eyJraaBcDeF1234567890&access_token=eyJraGhIjKlM1112131415&expires_in=3600&token_type=
```

Dettagli sull'interfaccia utente ospitata dei pool di utenti di Amazon Cognito

L'interfaccia utente ospitata e la conferma degli utenti come amministratore

Per gli utenti locali del pool di utenti, l'interfaccia utente ospitata funziona meglio quando si configura il pool di utenti su Consenti a Cognito di inviare automaticamente messaggi per la verifica e la conferma. Quando abiliti questa impostazione, Amazon Cognito invia un messaggio con un codice di conferma agli utenti che effettuano la registrazione. Quando invece confermi gli utenti come un amministratore del pool di utenti, l'interfaccia utente ospitata visualizza un messaggio di errore dopo la registrazione. In questo stato, Amazon Cognito ha creato il nuovo utente, ma non è stato in grado di inviare un messaggio di verifica. Puoi comunque confermare gli utenti come un amministratore, ma potrebbero contattare il supporto tecnico dopo che hanno rilevato un errore. Per ulteriori informazioni sulla conferma amministrativa, consulta Permettere agli utenti di registrarsi ma confermarli come un amministratore del pool di utenti.

Visualizzazione delle modifiche alla configurazione dell'interfaccia utente ospitata

Se le modifiche alle pagine dell'interfaccia utente ospitata non vengono visualizzate immediatamente, attendi qualche minuto, quindi aggiorna la pagina.

Decodifica dei token del pool di utenti

I token dei pool di utenti di Amazon Cognito vengono firmati utilizzando un algoritmo RS256. Puoi decodificare e verificare i token del pool di utenti utilizzando AWS Lambda, vedi Decodificare e verificare i token Amazon Cognito JWT su. GitHub

L'interfaccia utente e la versione TLS ospitate

L'interfaccia utente ospitata richiede la crittografia in transito. I domini del pool di utenti forniti da Amazon Cognito richiedono una versione TLS minima di 1.2. I domini personalizzati supportano ma non richiedono la versione TLS 1.2. Poiché Amazon Cognito gestisce la configurazione dell'interfaccia utente ospitata e degli endpoint del server di autorizzazione, non puoi modificare i requisiti TLS del dominio del tuo pool di utenti.

L'interfaccia utente ospitata e le policy CORS

L'interfaccia utente ospitata di Amazon Cognito non supporta le policy di condivisione di risorse tra origini (CORS) personalizzate. Una policy CORS nell'interfaccia utente ospitata impedirebbe agli utenti di passare parametri di autenticazione nelle loro richieste. Implementa invece una policy CORS nel front-end Web della tua app. Amazon Cognito restituisce un'intestazione di risposta `Access-Control-Allow-Origin: *` per richieste ai seguenti endpoint OAuth.

1. Endpoint Token
2. Endpoint Revoke
3. Endpoint UserInfo

Cookie dell'interfaccia utente e del server di autorizzazione ospitati

Gli endpoint del pool di utenti di Amazon Cognito impostano i cookie nei browser degli utenti. I cookie sono conformi ai requisiti di alcuni browser in base ai quali i siti non impostano cookie di terze parti. Sono limitati solo agli endpoint del pool di utenti e includono quanto segue:

- Un `XSRF-TOKEN` cookie per ogni richiesta.
- Un `csrf-state` cookie per la coerenza della sessione quando un utente viene reindirizzato.
- Un cookie `cognito` di sessione che conserva i tentativi di accesso riusciti per un'ora.

Configurazione di un dominio di bacino d'utenza

Dopo aver configurato un client dell'App, puoi configurare l'indirizzo delle pagine Web per la registrazione e l'accesso. Puoi utilizzare un dominio ospitato Amazon Cognito e scegliere un prefisso di dominio disponibile oppure puoi utilizzare il tuo indirizzo Web come dominio personalizzato.

Per aggiungere un client di applicazioni e un dominio ospitato di Amazon Cognito con la AWS Management Console, consulta la sezione Aggiunta di un'app per abilitare l'interfaccia utente Web ospitata.

> **Note**
>
> Non è possibile utilizzare il testo `aws`, `amazon` o `cognito` nel prefisso del dominio.

Argomenti

- Utilizzo del dominio di Amazon Cognito per l'interfaccia utente ospitata
- Utilizzo del proprio dominio per l'interfaccia utente ospitata

Utilizzo del dominio di Amazon Cognito per l'interfaccia utente ospitata

Dopo aver configurato un client dell'App, puoi configurare l'indirizzo delle pagine Web per la registrazione e l'accesso. È possibile utilizzare il dominio ospitato Amazon Cognito con il prefisso del tuo dominio.

> **Note**
>
> Per aumentare la sicurezza delle tue applicazioni Amazon Cognito, i domini principali degli endpoint del pool di utenti sono registrati nella Public Suffix List (PSL). La PSL aiuta i browser Web degli utenti a comprendere in modo coerente gli endpoint del pool di utenti e i cookie da essi impostati.
> I domini principali degli endpoint del pool di utenti hanno i seguenti formati.
>
> ```
> auth.Region.amazoncognito.com
> auth-fips.Region.amazoncognito.com
> ```

Per aggiungere un client per l'app e un dominio ospitato su Amazon Cognito con AWS Management Console, consulta. Creazione di un client dell'app

Argomenti

- Prerequisiti
- Fase 1: Configurazione di un dominio del bacino d'utenza ospitato
- Fase 2: Verifica della pagina di accesso

Prerequisiti

Prima di iniziare è necessario disporre di quanto segue:

- Un bacino d'utenza con un client dell'App. Per ulteriori informazioni, consulta Nozioni di base sui bacini d'utenza.

Fase 1: Configurazione di un dominio del bacino d'utenza ospitato

Per configurare un dominio del bacino d'utenza ospitato.

Puoi utilizzare l'API o l' AWS Management Console AWS CLI o per configurare un dominio con pool di utenti.

Amazon Cognito console

Configura un dominio

1. Passa alla scheda App Integration (Integrazione App) per il bacino d'utenza.
2. Accanto a Dominio, seleziona Operazioni, quindi seleziona Crea dominio personalizzato o Crea dominio Amazon Cognito. Se hai già configurato un dominio del pool di utenti, scegli Elimina dominio Amazon Cognito o Elimina dominio personalizzato prima di creare il nuovo dominio personalizzato.
3. Inserisci un prefisso dominio disponibile da utilizzare con un Dominio Amazon Cognito. Per informazioni sulla configurazione di un Dominio personalizzato, consulta la sezione Utilizzo del proprio dominio per l'interfaccia utente ospitata
4. Scegli Create (Crea).

CLI/API

Utilizza i seguenti comandi per creare un prefisso di dominio e assegnarlo al tuo bacino d'utenza.

Per configurare un dominio del bacino d'utenza

- AWS CLI: `aws cognito-idp create-user-pool-domain`

  Esempio: `aws cognito-idp create-user-pool-domain --user-pool-id <user_pool_id> --domain <domain_name>`
- AWS API: CreateUserPoolDomain

Ottenimento di informazioni su un dominio

- AWS CLI: `aws cognito-idp describe-user-pool-domain`

  Esempio: `aws cognito-idp describe-user-pool-domain --domain <domain_name>`
- AWS API: DescribeUserPoolDomain

Eliminazione di un dominio

- AWS CLI: `aws cognito-idp delete-user-pool-domain`

  Esempio: `aws cognito-idp delete-user-pool-domain --domain <domain_name>`
- AWS API: DeleteUserPoolDomain

Fase 2: Verifica della pagina di accesso

- Verifica che la pagina di accesso sia disponibile dal tuo dominio ospitato di Amazon Cognito.

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

Il dominio viene visualizzato nella pagina Nome dominio della console Amazon Cognito. L'ID del client app e l'URL di callback sono visualizzati nella pagina App client settings (Impostazioni client dell'app).

Utilizzo del proprio dominio per l'interfaccia utente ospitata

Dopo aver impostato un client di app, puoi configurare il bacino d'utenza con un dominio personalizzato per l'interfaccia utente ospitata di Amazon Cognito e gli endpoint dell'API auth. Il dominio personalizzato consente agli utenti di accedere all'applicazione utilizzando il tuo indirizzo Web.

Argomenti

- Aggiunta di un dominio personalizzato a un bacino d'utenza
- Modifica del certificato SSL per il dominio personalizzato

Aggiunta di un dominio personalizzato a un bacino d'utenza

Per aggiungere un dominio personalizzato al bacino d'utenza, devi specificare il nome di dominio nella console Amazon Cognito e fornire un certificato da gestire con AWS Certificate Manager (ACM). Dopo avere aggiunto il dominio, Amazon Cognito fornisce una destinazione alias che devi aggiungere alla configurazione DNS.

Prerequisiti

Prima di iniziare è necessario disporre di quanto segue:

- Un bacino d'utenza con un client dell'App. Per ulteriori informazioni, consulta Nozioni di base sui bacini d'utenza.
- Un dominio Web di tua proprietà, Il relativo dominio padre deve avere un record A DNS valido. Puoi assegnare qualsiasi valore a questo record. Il padre può essere la root del dominio o un dominio figlio che è un gradino più in alto nella gerarchia dei domini. Ad esempio, se il dominio personalizzato è auth.xyz.example.com, Amazon Cognito deve poter risolvere xyz.example.com in un indirizzo IP. Per evitare ripercussioni accidentali sull'infrastruttura del cliente, Amazon Cognito non supporta l'uso di domini di primo livello (TLD) per i domini personalizzati. Per ulteriori informazioni, consulta la pagina relativa ai nomi di dominio.
- La possibilità di creare un dominio secondario per il dominio personalizzato. È consigliabile usare auth come sottodominio. Ad esempio: `auth.example.com`.

> **Note**
>
> Se non disponi di un certificato jolly, potresti dover ottenere un nuovo certificato per il dominio secondario del tuo dominio personalizzato.

- Un certificato Secure Sockets Layer (SSL) gestito da ACM.

> **Note**
>
> È necessario modificare la AWS regione in Stati Uniti orientali (Virginia settentrionale) nella console ACM prima di richiedere o importare un certificato.

- Un'applicazione che consente al server di autorizzazione del pool di utenti di aggiungere cookie alle sessioni utente. Amazon Cognito imposta diversi cookie necessari per l'interfaccia utente ospitata. Tra queste vi sono `cognito`, `cognito-fl` e `XSRF-TOKEN`. Sebbene ogni singolo cookie sia conforme ai limiti di dimensione del browser, le modifiche alla configurazione del pool di utenti potrebbero causare un aumento delle dimensioni dei cookie dell'interfaccia utente ospitata. Un servizio intermedio come un Application Load Balancer (ALB) davanti al tuo dominio personalizzato potrebbe imporre una dimensione massima dell'intestazione o una dimensione totale dei cookie. Se l'applicazione imposta anche i propri cookie, le sessioni degli utenti potrebbero superare questi limiti. Per evitare conflitti relativi ai limiti di dimensione, consigliamo all'applicazione di non impostare i cookie nel sottodominio dell'interfaccia utente ospitato.
- Autorizzazione ad aggiornare CloudFront le distribuzioni Amazon. A tale scopo, puoi associare la dichiarazione di policy IAM seguente a un utente nel tuo Account AWS:

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Sid": "AllowCloudFrontUpdateDistribution",
            "Effect": "Allow",
            "Action": [
                "cloudfront:updateDistribution"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
```

```
}
```

Per ulteriori informazioni sull'autorizzazione delle azioni in CloudFront, consulta Using Identity-Based Policies (IAM Policies) per. CloudFront

Amazon Cognito inizialmente utilizza le tue autorizzazioni IAM per configurare la CloudFront distribuzione, ma la distribuzione è gestita da. AWS Non puoi modificare la configurazione della CloudFront distribuzione associata da Amazon Cognito al tuo pool di utenti. Ad esempio, non è possibile aggiornare le versioni TLS supportate nella policy di sicurezza.

Fase 1: Inserimento del nome di dominio personalizzato

È possibile aggiungere il dominio al bacino d'utenza utilizzando la console o l'API Amazon Cognito.

Amazon Cognito console

Aggiungere il dominio al bacino d'utenza dalla console Amazon Cognito:

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli Bacini d'utenza.
3. Scegli il bacino d'utenza da aggiungere al dominio.
4. Scegli la scheda App integration (Integrazione app).
5. Accanto a Dominio, scegli Operazioni, quindi scegli Create custom domain (Crea un dominio personalizzato).

   > **Note**
   >
   > Se hai già configurato un dominio del bacino d'utenza, scegli Delete Cognito domain (Elimina dominio Cognito) o Delete custom domain (Elimina dominio personalizzato) per eliminare il dominio esistente prima di creare il nuovo dominio personalizzato.

6. Alla voce Custom domain (Dominio personalizzato), inserisci l'URL del dominio che vuoi utilizzare con Amazon Cognito. Il nome di dominio può contenere solo lettere minuscole, numeri e trattini. Non utilizzare un trattino come primo o ultimo carattere. Utilizzare i punti per separare i nomi di dominio secondario.
7. Alla voce ACM certificate (Certificato ACM), scegli il certificato SSL che desideri utilizzare per il tuo dominio. Solo i certificati ACM negli Stati Uniti orientali (Virginia settentrionale) possono

essere utilizzati con un dominio personalizzato Amazon Cognito, indipendentemente dal pool di Regione AWS utenti.

Se non disponi di un certificato, è possibile utilizzare ACM per effettuare il provisioning di uno negli Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni, consulta Nozioni di base nella Guida per l'utente di AWS Certificate Manager .

8. Scegli Crea.
9. Amazon Cognito ti reindirizzerà alla scheda App integration (Integrazione app). Viene visualizzato un messaggio Create an alias record in your domain's DNS (Crea un registro di alias nel DNS del tuo dominio). Prendi nota del Domain (Dominio) e della Alias target (Destinazione alias) visualizzati nella console. Saranno utilizzati nella fase successiva per indirizzare il traffico verso il dominio personalizzato.

API

Aggiungere il dominio al bacino d'utenza con API Amazon Cognito:

- Usa l'azione CreateUserPoolDomain.

Fase 2: Aggiunta di una destinazione alias e di sottodomini

In questa fase, è possibile configurare, mediante il fornitore di servizi DNS (Domain Name Server), un alias che punti alla destinazione di alias della fase precedente. Se per la risoluzione dell'indirizzo DNS utilizzi Amazon Route 53, passa alla sezione Come aggiungere una destinazione alias e un sottodominio utilizzando Route 53.

Per aggiungere una destinazione alias e un sottodominio all'attuale configurazione DNS

- Se non utilizzi Route 53 per la risoluzione dell'indirizzo DNS, dovrai utilizzare gli strumenti di configurazione del tuo provider di servizi DNS per aggiungere la destinazione alias della fase precedente al registro DNS del tuo dominio. Il fornitore DNS dovrà inoltre configurare il sottodominio per il dominio personalizzato.

Come aggiungere una destinazione alias e un sottodominio utilizzando Route 53

1. Accedi alla console Route 53. Se richiesto, inserisci le credenziali AWS .

2. Se non disponi di una zona ospitata in Route 53, creane una con una radice che sia l'elemento principale del tuo dominio personalizzato. Per ulteriori informazioni, consulta la pagina

   a. Scegli Create Hosted Zone (Crea zona ospitata).

   b. Inserisci il dominio principale, ad esempio *auth.example.com*, del tuo dominio personalizzato, ad esempio *myapp.auth.example.com*, dall'elenco Domain Name (Nome dominio).

   c. Inserisci una Descrizione per la zona ospitata.

   d. Scegli un Tipo di zona ospitata di una Zona ospitata pubblica per consentire ai client pubblici di risolvere il tuo dominio personalizzato. La scelta della Zona ospitata privata non è supportata.

   e. Applica Tag come vuoi.

   f. Scegli Crea zona ospitata.

   > **Note**
   >
   > È inoltre possibile creare una nuova zona ospitata per il dominio personalizzato e creare un set di deleghe nella zona ospitata principale che indirizza le query alla zona ospitata del sottodominio. Altrimenti, crea un record A. Questo metodo offre maggiore flessibilità e sicurezza con le zone ospitate. Per ulteriori informazioni, consulta la sezione Creating a subdomain for a domain hosted through Amazon Route 53 (Creazione di un sottodominio per un dominio in hosting tramite Amazon Route 53).

3. Nella pagina Hosted Zones (Zone ospitate), scegli il nome della tua zona ospitata.

4. Aggiungi un record DNS per il dominio principale del tuo dominio personalizzato, se non ne hai già uno. Aggiungi un `A` record DNS per il dominio principale e scegli Crea record. Di seguito è riportato un registro di esempio per il dominio *auth.example.com*.

   ```
   auth.example.com. 60 IN A 198.51.100.1
   ```

   > **Note**
   >
   > Amazon Cognito verifica che sia presente un registro DNS per il dominio principale del tuo dominio personalizzato per proteggerti dal dirottamento accidentale dei domini di produzione. Se non disponi di un registro DNS per il dominio principale, Amazon Cognito restituirà un errore quando si tenta di impostare il dominio personalizzato. Un record

Start of Authority (SOA) non è un record DNS sufficiente ai fini della verifica del dominio principale.

5. Aggiungi un record DNS per il tuo dominio personalizzato. Il record deve puntare alla destinazione Alias del dominio personalizzato, ad esempio. `123example.cloudfront.net` Scegliere nuovamente Create record (Crea registro).
6. Inserire un nome di registro che corrisponde al tuo dominio personalizzato, ad esempio *`myapp`* per creare un registro per *`myapp.auth.example.com`*.
7. Abilitare l'opzione Alias.
8. In Route traffic to (Instradare il traffico a), scegliere Alias to CloudFront distribution (Distribuzione da Alias a Cloudfront). Inserisci la Destinazione alias fornita da Amazon Cognito quando hai creato il tuo dominio personalizzato.
9. Scegli Create Records (Crea registri).

> **Note**
>
> I nuovi registri possono richiedere circa 60 secondi per la propagazione a tutti i server DNS di Route 53. Puoi utilizzare il metodo dell'GetChangeAPI Route 53 per verificare che le modifiche si siano propagate.

Fase 3: Verifica della pagina di accesso

- Verifica che la pagina di accesso sia disponibile dal tuo dominio personalizzato.

  Accedi con il tuo dominio personalizzato e con il sottodominio immettendo questo indirizzo nel browser. Un esempio di un URL di dominio personalizzato è *`esempio.com`* con il sottodominio *`auth`*:

```
https://myapp.auth.example.com/login?
response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callback_url>
```

Modifica del certificato SSL per il dominio personalizzato

Se necessario, puoi utilizzare Amazon Cognito per modificare il certificato che hai applicato al tuo dominio personalizzato.

Di solito, questa operazione non è necessaria dopo la procedura di rinnovo dei certificati con ACM. Quando rinnovi il certificato esistente in ACM, l'ARN del tuo certificato rimane invariato e il tuo dominio personalizzato utilizza automaticamente il nuovo certificato.

Tuttavia, se sostituisci il certificato esistente con uno nuovo, ACM assegna un nuovo ARN al nuovo certificato. Per applicare il nuovo certificato al dominio personalizzato, devi fornire questo ARN ad Amazon Cognito.

Dopo avere fornito il nuovo certificato, Amazon Cognito impiega fino a 1 ora per distribuirlo nel dominio personalizzato.

> **Prima di iniziare**
>
> Prima di poter modificare il certificato in Amazon Cognito, devi aggiungerlo ad ACM. Per ulteriori informazioni, consulta Nozioni di base nella Guida per l'utente di AWS Certificate Manager .
> Quando aggiungi il certificato ad ACM, devi scegliere Stati Uniti orientali (Virginia settentrionale) come regione AWS .

Puoi modificare il tuo certificato utilizzando la console o l'API Amazon Cognito.

AWS Management Console

Rinnovare un certificato dalla console Amazon Cognito:

1. Accedi AWS Management Console e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito/home
2. Scegli Bacini d'utenza.
3. Scegli il bacino d'utenza per cui desideri aggiornare il certificato.
4. Scegli la scheda App integration (Integrazione app).
5. Scegli Operazioni, Edit ACM certificate (Modifica del certificato ACM.
6. Scegli il nuovo certificato che desideri associare al dominio personalizzato.
7. Scegli Save changes (Salva modifiche).

API

Come rinnovare un certificato (API Amazon Cognito)

- Usa l'azione UpdateUserPoolDomain.

Personalizzazione delle pagine Web di registrazione e accesso integrate

È possibile utilizzare la AWS Management Console o la AWS CLI o l'API, per specificare le impostazioni di personalizzazione per l'interfaccia utente delle app integrate. È possibile caricare un'immagine del logo personalizzata da visualizzare nell'app. È inoltre possibile utilizzare fogli di stile Cascading (CSS) per personalizzare l'aspetto dell'interfaccia utente.

È possibile specificare le impostazioni di personalizzazione interfaccia utente dell'App per un singolo client (con uno specifico `clientId`) o per tutti i clienti (impostando il `clientId` su `ALL`). Se specifichi `ALL`, la configurazione di default sarà usata per ogni client che non ha nessuna personalizzazione dell'interfaccia utente impostata in precedenza. Se specifichi le impostazioni di personalizzazione dell'interfaccia utente per un determinato client, non sarà più possibile ripristinare la configurazione `ALL`.

Le dimensioni della richiesta che imposta la personalizzazione dell'interfaccia utente non devono essere superiori a 135 KB. In rari casi, le dimensioni della somma delle intestazioni di richiesta, del file CSS e del logo potrebbero eccedere i 135 KB. Amazon Cognito codifica il file di immagine in Base64. Ciò aumenta le dimensioni di un'immagine da 100 KB a 130 KB, mantenendo cinque KB per intestazioni della richiesta e CSS. Se le dimensioni della richiesta sono troppo elevate, la AWS Management Console o la richiesta API `SetUICustomization` restituisce un errore `request parameters too large`. Regolare l'immagine del logo in modo che le dimensioni non superino i 100 KB e le dimensioni del file CSS non siano maggiori di 3 KB. Non è possibile impostare CSS e personalizzazione del logo separatamente.

> **Note**
>
> Per personalizzare l'interfaccia utente, è necessario configurare un dominio per il pool di utenti.

Specificazione di un logo personalizzato per l'App

Amazon Cognito centra il logo personalizzato sopra i campi di input in Endpoint Login.

Scegli un file PNG, JPG o JPEG scalabile fino a 350 x 178 pixel per il logo personalizzato dell'interfaccia utente ospitata. Le dimensioni del file del logo non possono essere maggiori di 100 KB o 130 KB dopo la codifica di Amazon Cognito in Base64. Per impostare un `ImageFile` in SetUICustomization nell'API, converti il file in una stringa di testo con codifica Base64 oppure, nella AWS CLI, fornisci un percorso file e lascia che venga codificato da Amazon Cognito.

Specificazione delle personalizzazioni CSS per l'App

È possibile personalizzare il CSS per le pagine dell'app ospitata, con le seguenti restrizioni:

- È possibile utilizzare uno dei seguenti nomi di classe CSS:
  - `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`
- I valori delle proprietà possono contenere HTML, ad eccezione dei seguenti valori: `@import`, `@supports`, `@page`, oppure istruzioni `@media` o Javascript.

È possibile personalizzare le seguenti proprietà CSS.

Etichette

- font-weight è un multiplo di 100 da 100 e 900.

Campi di input

- width è la larghezza in percentuale del blocco contenitore.
- height (altezza) è l'altezza del campo di input in pixel (px).
- color (colore) è il colore del testo. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del campo di input. Può essere un qualsiasi valore di colore dello standard CSS.
- border (bordo) è un valore dello standard CSS del bordo che specifica la larghezza, la trasparenza e il colore del bordo della finestra della tua app. Larghezza può essere qualsiasi valore da 1px a 100px. Trasparenza può essere solida o nessuna. Colore può essere un qualsiasi valore di colori standard.

Descrizioni testuali

- padding-top è la quantità di padding sopra la descrizione testuale.
- padding-bottom è la quantità di padding sotto la descrizione testuale.
- display può essere `block` o `inline`.
- font-size è la dimensione del font per le descrizioni testuali.

Pulsante di invio

- font-size è la dimensione del font del testo del pulsante
- font-weight è il peso del font del testo del pulsante `bold`, `italic` o `normal`.
- margine è una stringa di quattro valori che indicano le dimensioni dei margini superiore, destro, inferiore e sinistro del pulsante.
- font-size è la dimensione del font per le descrizioni testuali.
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- height (altezza) è l'altezza del pulsante in pixel (px).
- color (colore) è il colore del testo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante. Può essere un valore di qualsiasi standard di colori.

Banner

- spaziatura interna è una stringa di quattro valori che indicano le dimensioni della spaziatura interna superiore, destro, inferiore e sinistro del banner.

- background-color è il colore di sfondo del banner. Può essere un qualsiasi valore di colore dello standard CSS.

Pulsante di invio con animazione al passaggio del mouse

- color (colore) è il colore di primo piano del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.

Pulsante provider di identità con animazione al passaggio del mouse

- color (colore) è il colore di primo piano del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante al passaggio del mouse. Può essere un qualsiasi valore di colore dello standard CSS.

Controllo password non valido

- color (colore) è il colore del testo del messaggio `"Password check not valid"`. Può essere un qualsiasi valore di colore dello standard CSS.

Contesto

- background-color è il colore di sfondo della finestra dell'app. Può essere un qualsiasi valore di colore dello standard CSS.

Messaggi di errore

- margine è una stringa di quattro valori che indicano le dimensioni dei margini superiore, destro, inferiore e sinistro.
- padding è la dimensione del padding.
- font-size è la dimensione del font.
- width (larghezza) è la larghezza del messaggio di errore in percentuale del blocco contenitore.
- background è il colore di sfondo del messaggio di errore. Può essere un qualsiasi valore di colore dello standard CSS.
- bordo è una stringa di tre valori che specificano la larghezza, trasparenza e colore del bordo.
- color (colore) è il colore del messaggio di errore. Può essere un qualsiasi valore di colore dello standard CSS.
- box-sizing viene utilizzata per indicare al browser quello che devono includere le proprietà di ridimensionamento (larghezza e altezza).

Pulsante provider di identità

- height (altezza) è l'altezza del pulsante in pixel (px).
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- text-align è l'impostazione per allineare il testo. Può essere `left`, `right` o `center`.
- margin-bottom è l'impostazione del margine inferiore.
- color (colore) è il colore del testo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- background-color è il colore di sfondo del pulsante. Può essere un qualsiasi valore di colore dello standard CSS.
- border-color è il colore di sfondo del bordo. Può essere un qualsiasi valore di colore dello standard CSS.

Descrizioni provider di identità

- padding-top è la quantità di padding sopra la descrizione.
- padding-bottom è la quantità di padding sotto la descrizione.
- display può essere `block` o `inline`.
- font-size è la dimensione del font per le descrizioni.

Testo legale

- color (colore) è il colore del testo. Può essere un qualsiasi valore di colore dello standard CSS.
- font-size è la dimensione del font.

> **Note**
>
> Personalizzando un Legal text (Testo legale), stai personalizzando il messaggio We won't post to any of your accounts without asking first (Non pubblicheremo su nessuno dei tuoi account senza il tuo consenso) che viene visualizzato sotto i provider di identità sociale nella pagina di accesso.

Logo

- max-width è la larghezza massima in percentuale del blocco contenitore.
- max-height è l'altezza massima in percentuale del blocco contenitore.

Focus del campo di input

- border-color è il colore del campo di input. Può essere un qualsiasi valore di colore dello standard CSS.
- outline è il bordo del campo di input in pixel.

Pulsante social

- height (altezza) è l'altezza del pulsante in pixel (px).
- text-align è l'impostazione per allineare il testo. Può essere `left`, `right` o `center`.
- width (larghezza) è la larghezza del testo del pulsante in percentuale del blocco contenitore.
- margin-bottom è l'impostazione del margine inferiore.

Controllo password valido

- color (colore) è il colore del testo del messaggio `"Password check valid"`. Può essere un qualsiasi valore di colore dello standard CSS.

Specifica delle impostazioni di personalizzazione dell'interfaccia utente dell'App per un bacino d'utenza (AWS Management Console)

È possibile usare la AWS Management Console per specificare le impostazioni di personalizzazione dell'interfaccia utente per la tua app.

> **Note**
>
> Puoi visualizzare l'interfaccia utente ospitata con le personalizzazioni costruendo il seguente URL, con le specifiche del bacino d'utenza, e inserendolo all'interno di un browser: `https://<your_domain>/login?response_type=code&client_id=<your_app_client_id>&redirect_uri=<your_callbac`
> Potrebbe essere necessario attendere fino a un minuto per aggiornare il browser prima che vengano visualizzate le modifiche effettuate nella console.
> Il dominio viene visualizzato nella scheda App integration (Integrazione app) alla voce Dominio. L'ID del client dell'app e l'URL di callback sono visualizzati nella scheda App client settings (Impostazioni client dell'app).

Specificazione delle impostazioni di personalizzazione dell'interfaccia utente dell'App

1. Accedi alla console Amazon Cognito.

2. Nel pannello di navigazione, scegli Bacini d'utenza e scegli i bacini d'utenza che intendi modificare.
3. Scegli la scheda Integrazione app.
4. Per personalizzare le impostazioni dell'interfaccia utente per tutti i client app, individua la voce Hosted UI customization (Personalizzazione dell'interfaccia utente) e seleziona Modifica.
5. Per personalizzare le impostazioni dell'interfaccia utente per un client dell'app, individua Client dell'app e seleziona il client dell'app che desideri modificare, quindi individua Personalizzazione dell'interfaccia utente ospitata e seleziona Modifica. Per passare a un client dell'App dalla personalizzazione predefinita del bacino d'utenza alla personalizzazione specifica del client, seleziona Use client-level settings (Utilizzo di impostazioni a livello di client).
6. Per caricare il tuo file immagine logo, scegli Choose file (Scegli file) o Replace current file (Sostituisci file attuale).
7. Per personalizzare il CSS per l'interfaccia utente ospitata, scarica CSS template.css e modifica il modello con i valori che desideri personalizzare. Solo le chiavi incluse nel modello possono essere utilizzate con l'interfaccia utente ospitata. Le chiavi CSS aggiunte non verranno applicate all'interfaccia utente. Dopo aver personalizzato il file CSS, scegli Choose file (Scegli file) o Replace current file (Sostituisci file corrente) per caricare il tuo CSS personalizzato.

Specifica delle impostazioni di personalizzazione dell'interfaccia utente dell'App per un bacino d'utenza (AWS CLI e API AWS)

Usa i seguenti comandi per specificare le impostazioni della personalizzazione dell'interfaccia utente dell'App del tuo bacino d'utenza.

Per ottenere le impostazioni di personalizzazione dell'interfaccia utente per l'interfaccia utente di un'app integrata del pool di utenti, utilizza le seguenti operazioni API.

- AWS CLI: `aws cognito-idp get-ui-customization`
- API AWS: GetUICustomization

Per configurare le impostazioni di personalizzazione dell'interfaccia utente per l'interfaccia utente di un'app integrata del pool di utenti, utilizza le seguenti operazioni API.

- AWS CLI dal file di immagine: `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 con immagine codificata come testo binario 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>;}"`
- API AWS: SetUICustomization

Registrazione e accesso con l'interfaccia utente ospitata

Dopo aver configurato e personalizzato l'interfaccia utente ospitata di Amazon Cognito per il pool di utenti e i client dell'app, l'app può presentarla agli utenti. L'interfaccia utente ospitata supporta diverse operazioni di autenticazione di Amazon Cognito, inclusi i seguenti esempi.

- Registrazione come nuovo utente IAM
- Verifica di un indirizzo e-mail o numero di telefono
- Configurazione dell'autenticazione a più fattori (MFA)
- Accesso con un nome utente e una password locali
- Accesso con un gestore dell'identità digitale (IdP) di terze parti
- Reimpostazione di una password

L'interfaccia utente ospitata di Amazon Cognito inizia in corrispondenza dell'Endpoint Login. L'URL della pagina di accesso è una combinazione del dominio scelto per il pool di utenti e dei parametri che riflettono le concessioni OAuth 2.0 da emettere, il client dell'app, il percorso dell'app e gli ambiti OpenID Connect (OIDC) da richiedere.

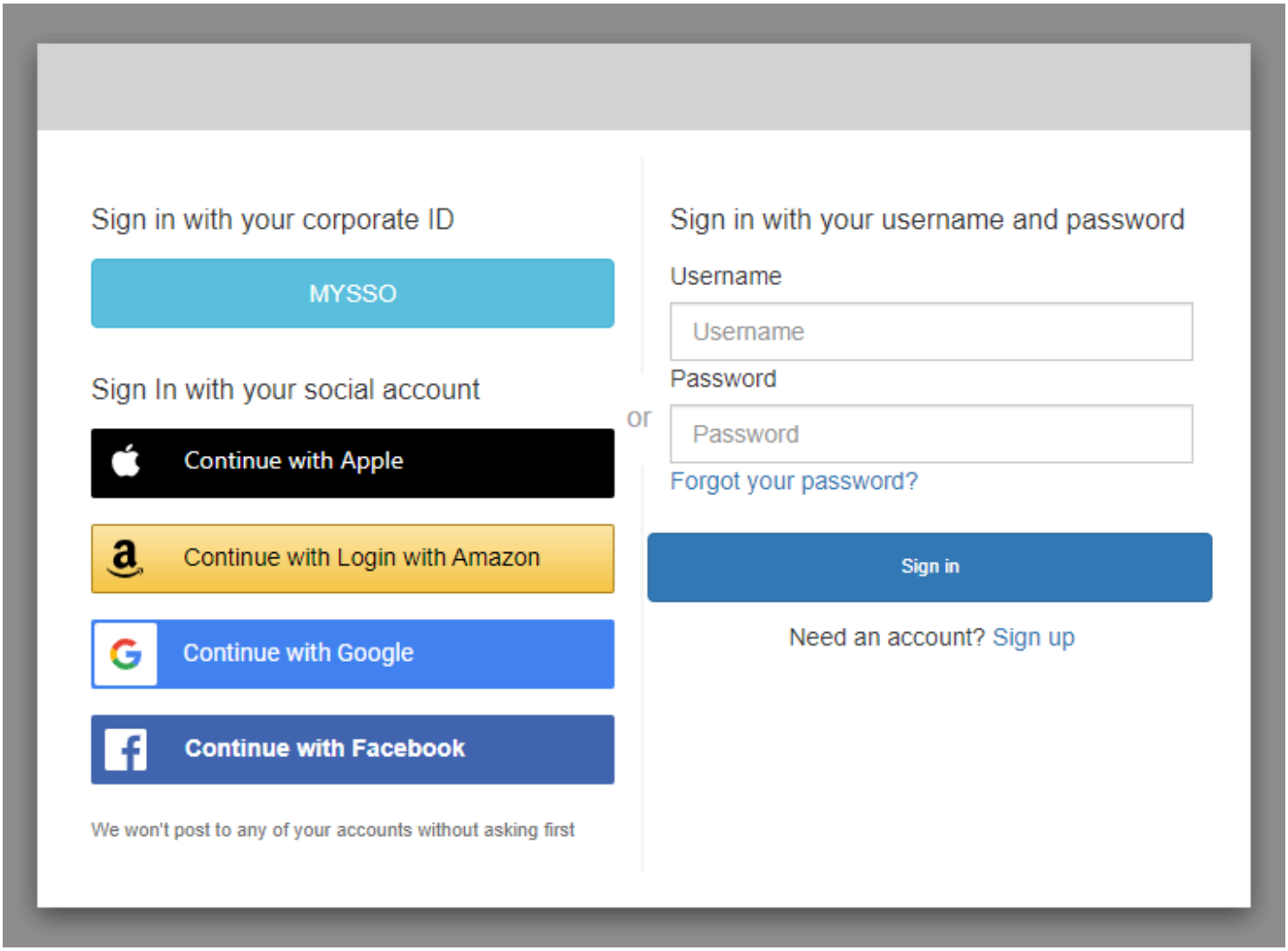
```
https://<your user pool domain>/authorize?client_id=<your app client
 ID>&response_type=<code/token>&scope=<scopes to request>&redirect_uri=<your callback
 URL>
```

Il seguente URL sostituisce i campi segnaposto riportati sopra con valori di esempio.

```
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 pagina di accesso per l'interfaccia utente ospitata di Amazon Cognito contiene opzioni per accedere tramite il pool di utenti o qualsiasi gestore dell'identità digitale (IdP) assegnato al client dell'app richiesto dall'utente. Include anche link per la registrazione di un nuovo account utente nel pool di utenti o per reimpostare una password dimenticata.



Argomenti

- Informazioni su come registrare un nuovo account nell'interfaccia utente ospitata di Amazon Cognito
- Informazioni su come effettuare l'accesso con l'interfaccia utente ospitata di Amazon Cognito
- Informazioni su come reimpostare una password con l'interfaccia utente ospitata di Amazon Cognito

Informazioni su come registrare un nuovo account nell'interfaccia utente ospitata di Amazon Cognito

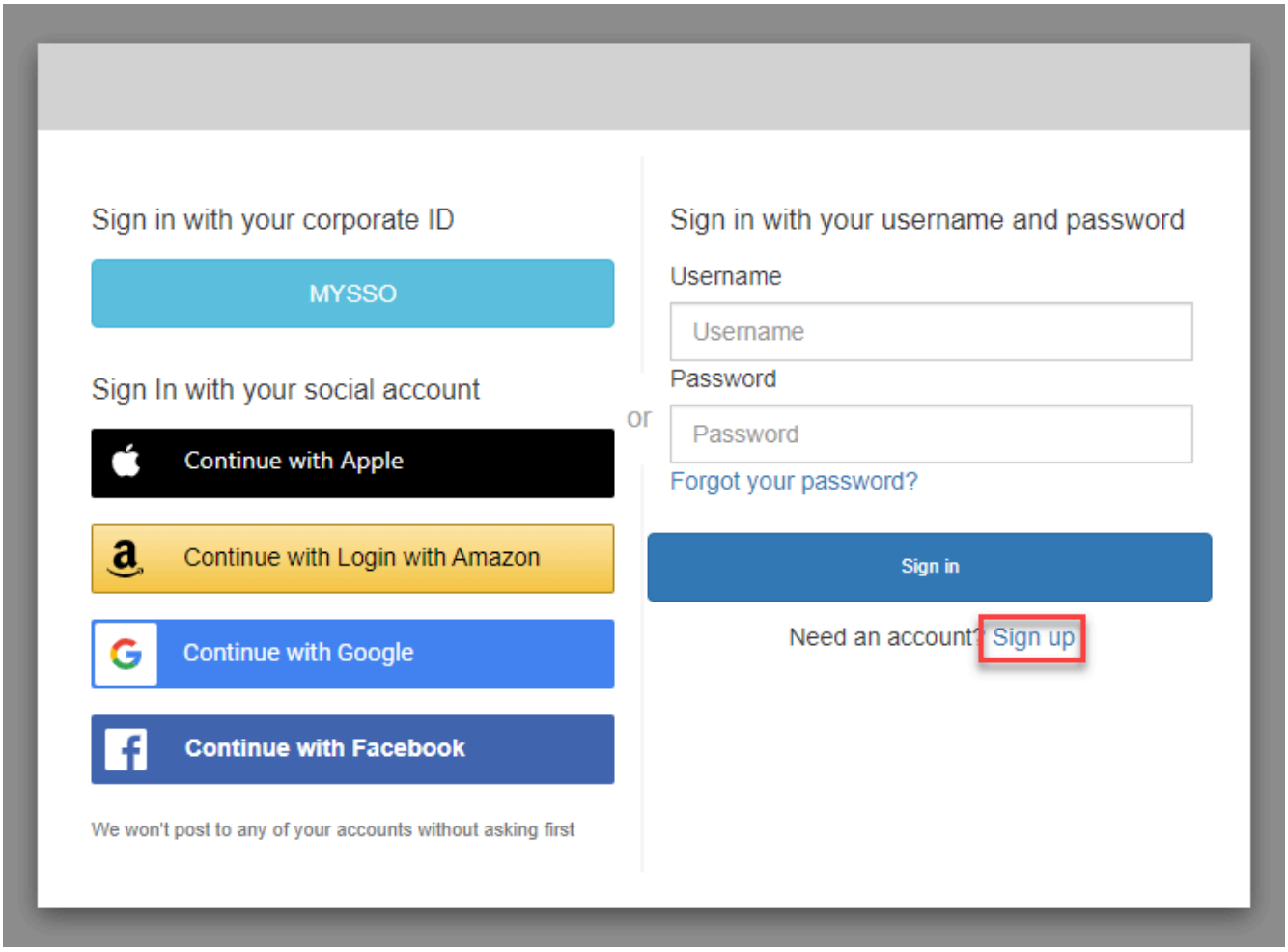
Questa guida mostra come registrare un account utente nelle app che utilizzano Amazon Cognito.

> **Note**
>
> Quando si accede a un'app che utilizza l'interfaccia utente ospitata di Amazon Cognito, è possibile che venga visualizzata una pagina personalizzata dal proprietario dell'app rispetto alla configurazione di base mostrata in questa guida.

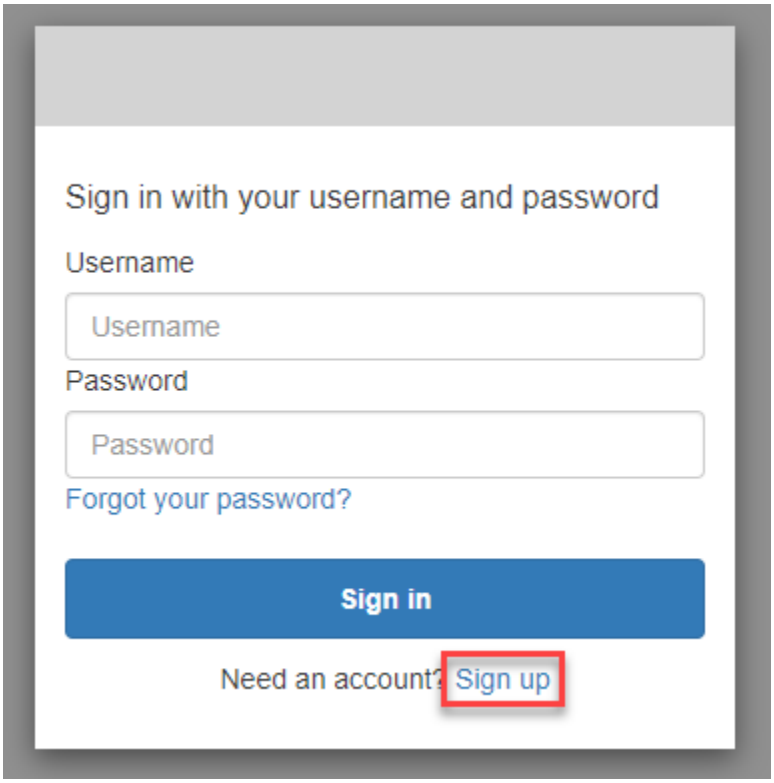
1. Scegli Sign up (Registra) nella pagina di accesso se intendi accedere tramite Amazon Cognito con un nome utente e una password, anziché una combinazione di nome utente e password dei provider di accesso di terze parti elencati dal proprietario dell'app.

   Se il provider di accesso non è Amazon Cognito, la registrazione è completa solo dopo aver scelto il pulsante del provider di terze parti. A seconda delle opzioni scelte dal proprietario dell'app, potrebbe venire visualizzato un elenco di provider da usare per l'accesso oppure essere necessario immettere un nome utente o una password.

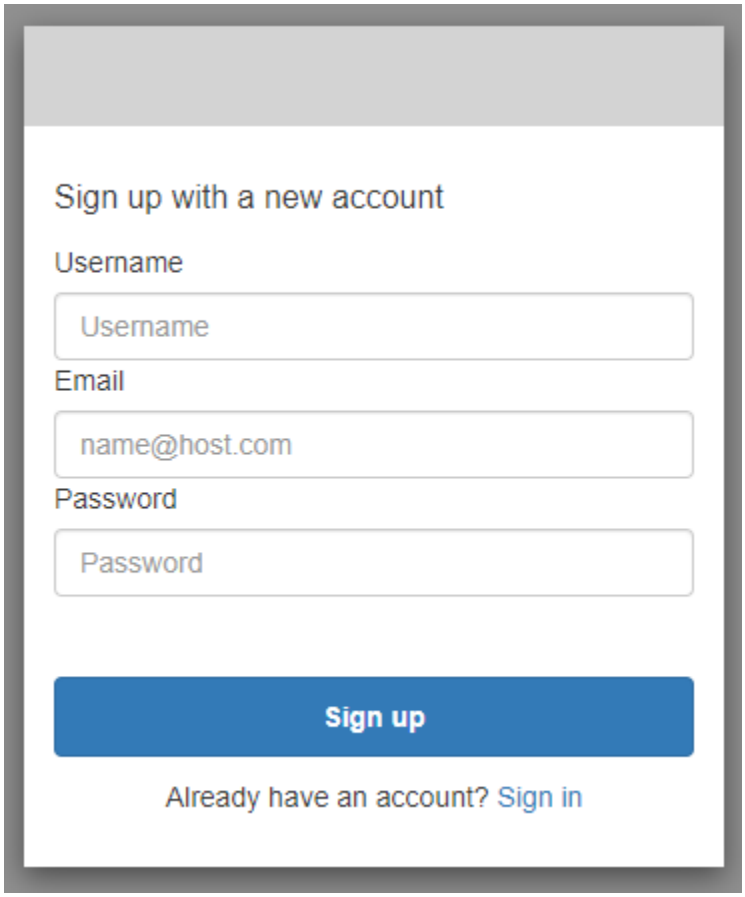
With multiple sign-in providers



With only Amazon Cognito as a sign-in provider

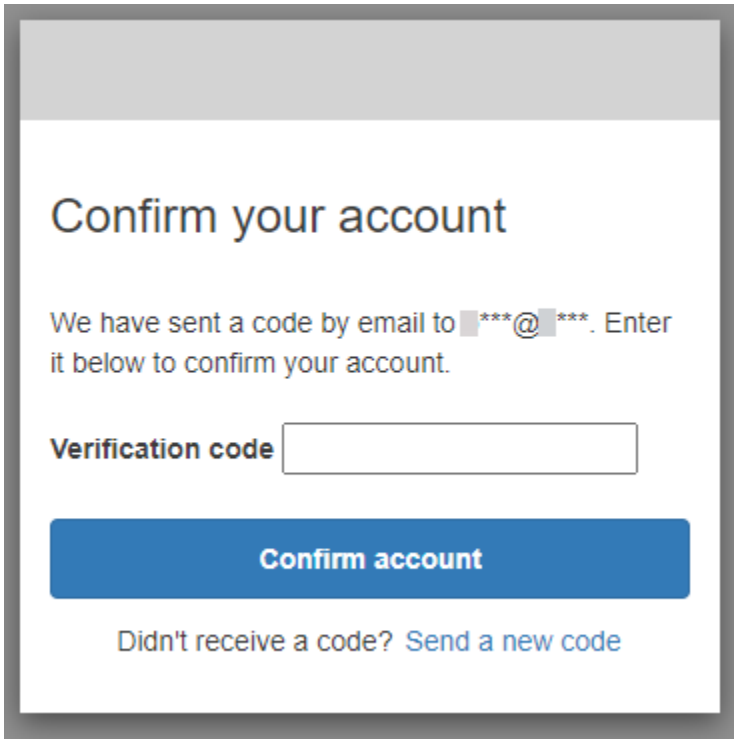


2. Nella pagina Sign up with a new account (Registrati con un nuovo account), il proprietario dell'app richiede le informazioni necessarie per registrarsi. Potrebbe venire richiesto un nome utente, un indirizzo e-mail o un numero di telefono. Immetti le informazioni richieste e scegli una password.



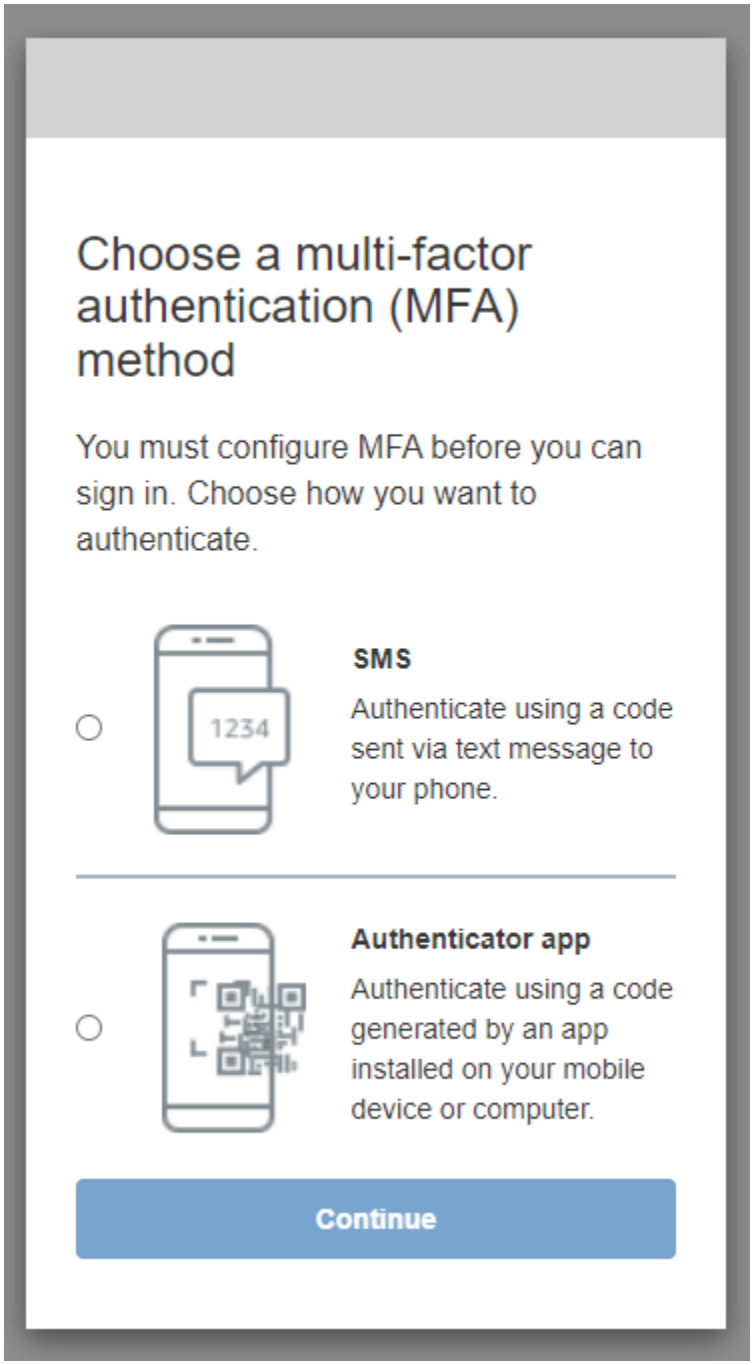
3. Nella pagina Confirm your account (Conferma account), il proprietario dell'app potrebbe avere impostato la richiesta di conferma dell'account per verificare che sia possibile ricevere messaggi all'indirizzo e-mail o al numero di telefono specificato.

   Riceverai un codice via e-mail o SMS. Inserisci il codice nel modulo per confermare di aver inserito le informazioni di contatto corrette.



4. Il proprietario dell'app potrebbe avere impostato la richiesta di configurazione dell'autenticazione a più fattori (MFA). È possibile che venga visualizzata la richiesta di scegliere il metodo MFA oppure l'app potrebbe passare alla fase successiva.

   Nella pagina Choose a multi-factor authentication (MFA) method (Scegli un metodo di autenticazione a più fattori [MFA]), scegli il metodo di autenticazione desiderato. Se scegli SMS, riceverai i passcode MFA mediante SMS. Se scegli Authenticator app (App Authenticator), è necessario installare un'app sul dispositivo per generare passcode MFA con scadenza. Devi effettuare una scelta entro 3 minuti.



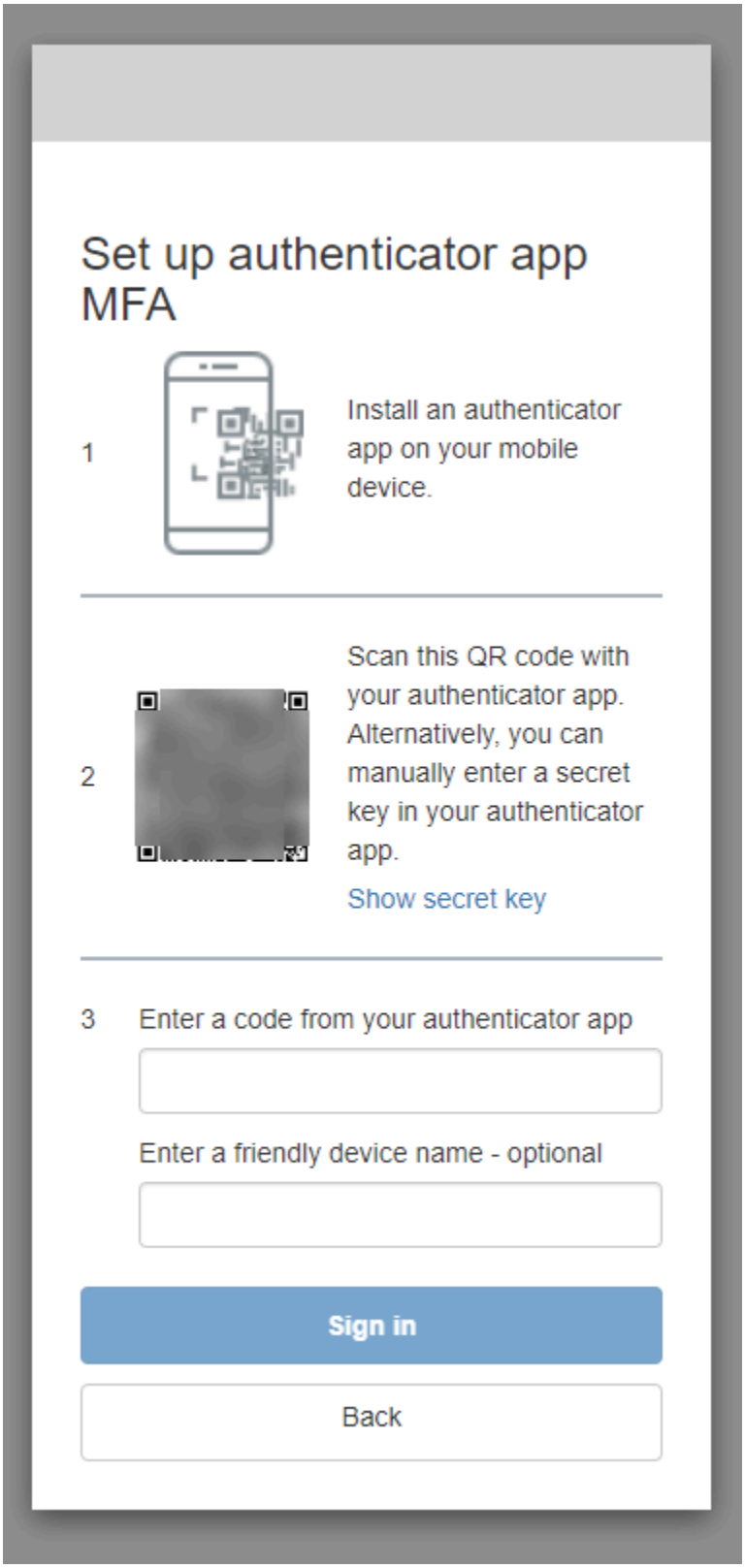
5. Amazon Cognito richiede l'immissione di un codice ricevuto dall'app Authenticator o mediante un SMS. Inserisci il codice ricevuto entro 3 minuti.

   Authenticator app

   1. Apri l'app Authenticator che hai scaricato.
   2. Scansiona il codice QR sulla pagina con la fotocamera. Potrebbe essere necessario autorizzare l'uso della fotocamera da parte dell'app.

Se non riesci a scansionare il codice QR, scegli Show secret key (Mostra chiave segreta) per visualizzare un codice che puoi inserire manualmente nell'app Authenticator.

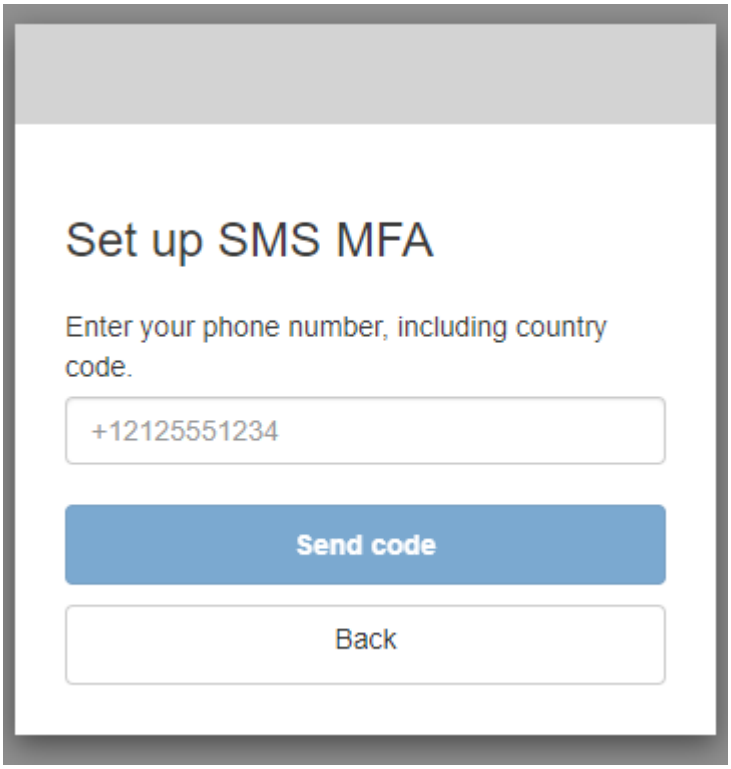
3. L'app Authenticator inizia a visualizzare codici che cambiano dopo pochi secondi. Inserisci un codice valido visualizzato nell'app.
4. (Facoltativo) Nella pagina Set up authenticator app MFA (Configura autenticazione a più fattori [MFA] per app Authenticator), scegli un nome per il tuo dispositivo. In fase di accesso, Amazon Cognito ti chiederà un codice inviato al dispositivo con il nome specificato in questo passaggio.


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
Back

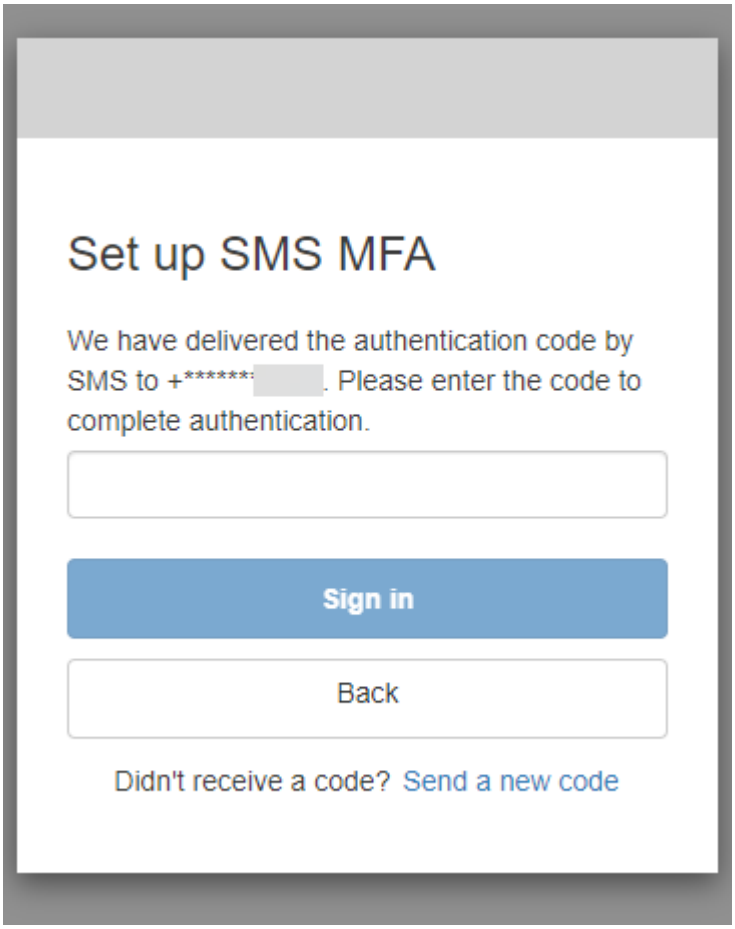
SMS text message

1. Se il proprietario dell'app non ha ancora acquisito il tuo numero di telefono, Amazon Cognito richiede il tuo numero di telefono.

   Nella pagina Set up SMS MFA (Configura MFA con SMS), inserisci un numero di telefono contenente il segno + e il prefisso internazionale, ad esempio +12125551234.



2. Riceverai un SMS contenente un codice. Nella pagina Set up SMS MFA (Configura MFA con SMS), inserisci il codice. Se non hai ricevuto il codice e vuoi riprovare, scegli Send a new code (Invia un nuovo codice). Seleziona Back (Indietro) per inserire un nuovo numero di telefono.



6. Alla prima registrazione e conferma dei dati, Amazon Cognito concede l'accesso all'app dopo aver completato questa procedura.

Informazioni su come effettuare l'accesso con l'interfaccia utente ospitata di Amazon Cognito

Questa guida mostra come accedere alle app che utilizzano Amazon Cognito.

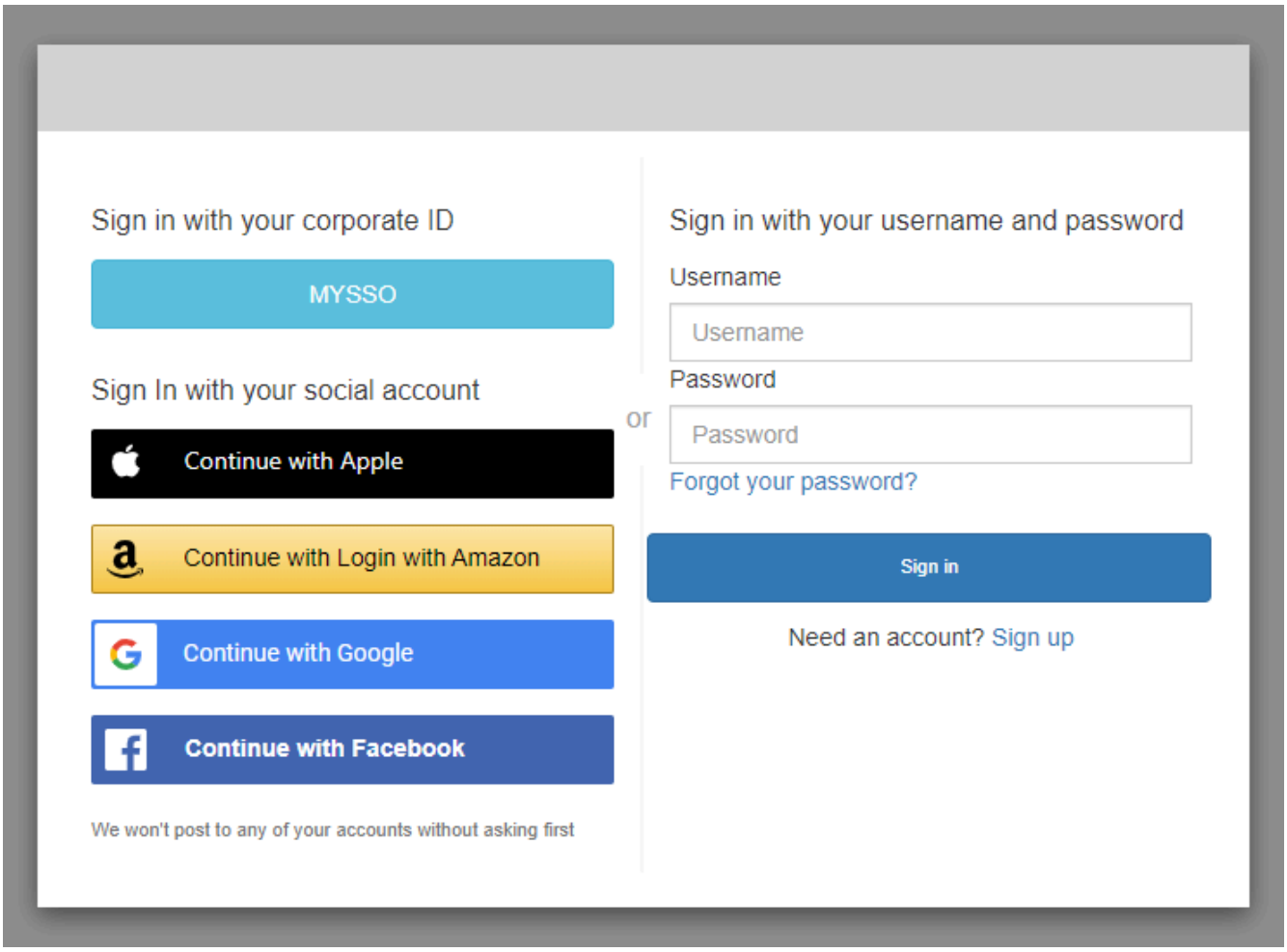
> **Note**
>
> Quando si accede a un'app che utilizza l'interfaccia utente ospitata di Amazon Cognito, è possibile che venga visualizzata una pagina personalizzata dal proprietario dell'app rispetto alla configurazione di base mostrata in questa guida.

1. A seconda delle opzioni scelte dal proprietario dell'app, potrebbe venire visualizzato un elenco di provider da usare per l'accesso oppure essere necessario immettere un nome utente o una

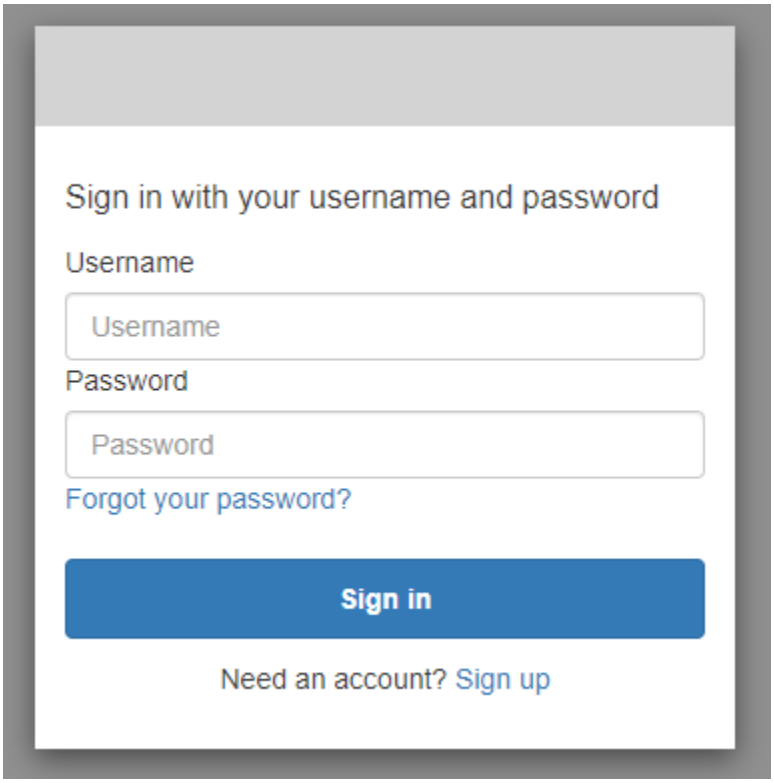
password. Quando si effettua l'accesso immettendo un nome utente e una password in questa pagina, significa che Amazon Cognito è il tuo provider di accesso. In caso contrario, il tuo provider di accesso corrisponde al pulsante che scegli.

Puoi scegliere un provider qui o inserire un nome utente e una password e accedere immediatamente alla tua app. Se Amazon Cognito è il tuo provider di accesso, il proprietario dell'app potrebbe anche avere impostato la richiesta di autenticazione a più fattori.

With multiple sign-in providers

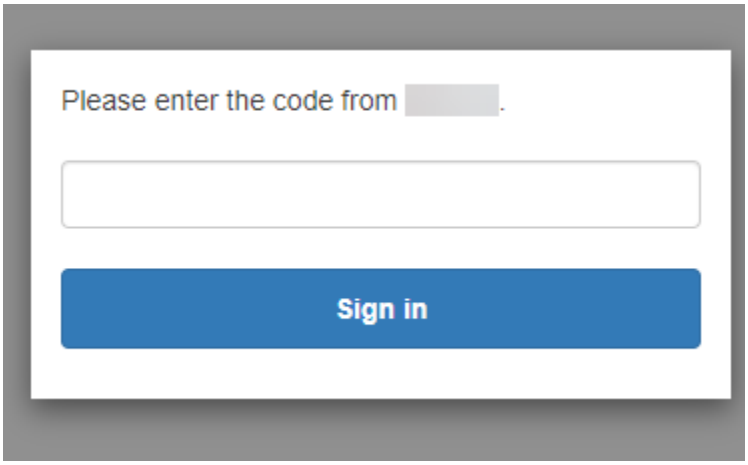


With only Amazon Cognito as a sign-in provider

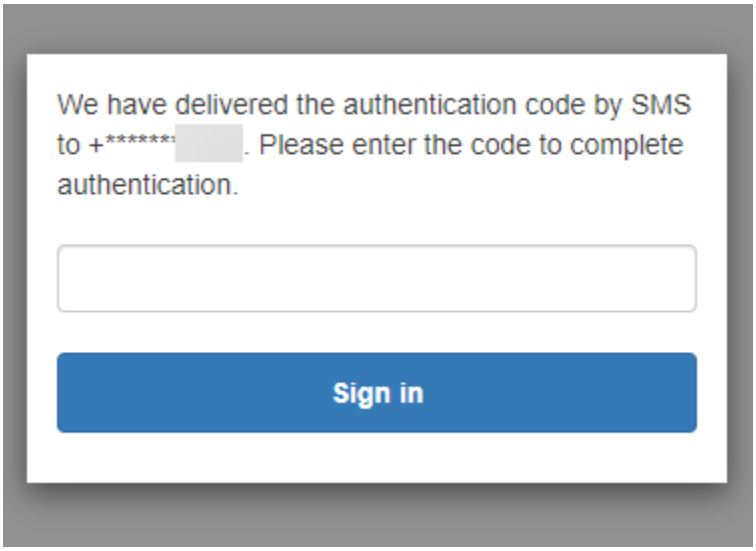


2. Potresti aver configurato l'autenticazione a più fattori (MFA) quando ti sei registrato nell'app. Inserisci il codice MFA che hai ricevuto via SMS o visualizzato nell'app Authenticator. Devi inserire questo codice entro 3 minuti.

   With an authenticator app



With an SMS code



3. Dopo aver effettuato l'accesso e aver completato l'autenticazione a più fattori (MFA), Amazon Cognito concede l'accesso alla tua app.

Informazioni su come reimpostare una password con l'interfaccia utente ospitata di Amazon Cognito

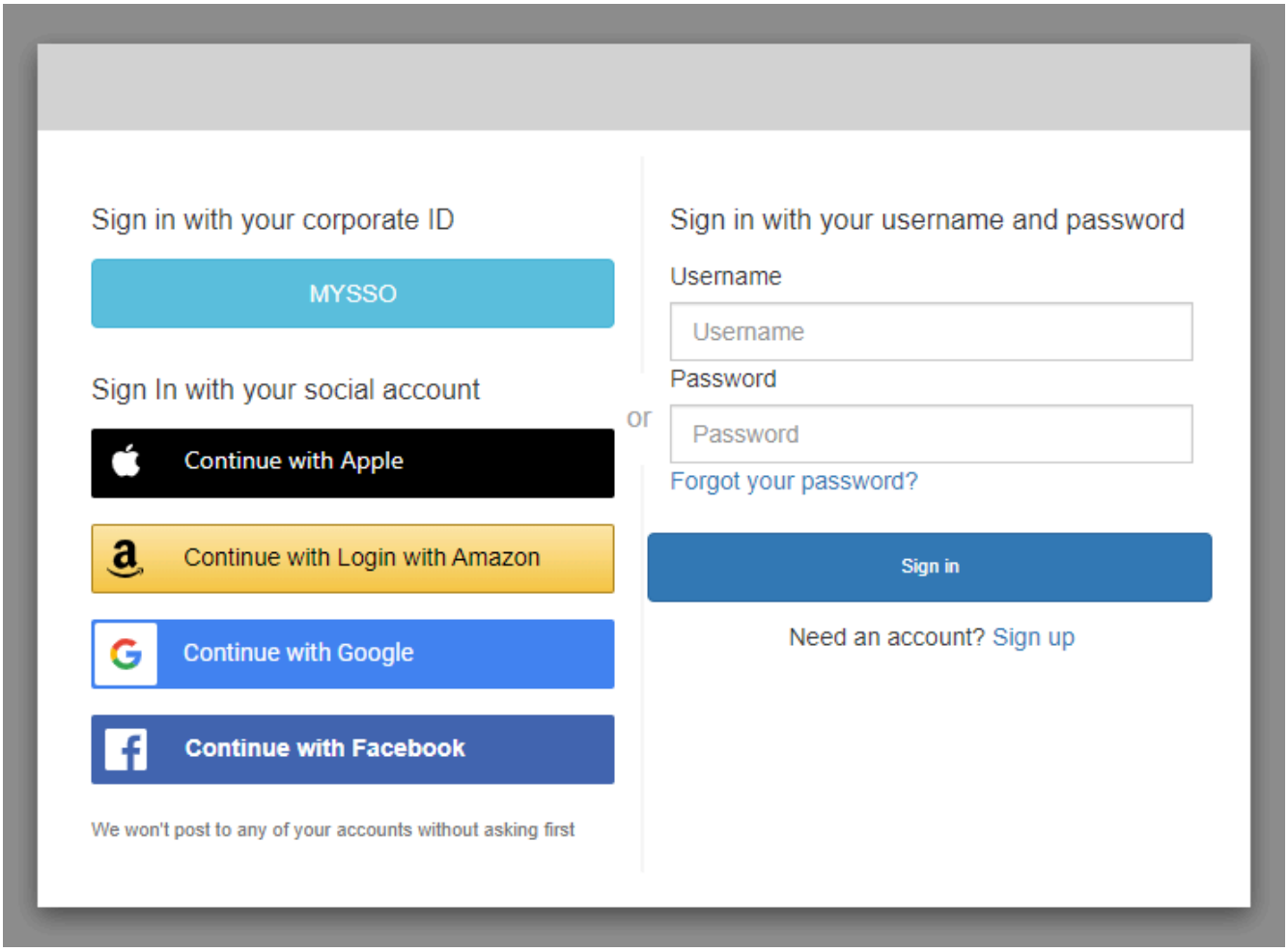
Questa guida mostra come reimpostare la password nelle app che utilizzano Amazon Cognito.

> **Note**
>
> Quando si accede a un'app che utilizza l'interfaccia utente ospitata di Amazon Cognito, è possibile che venga visualizzata una pagina personalizzata dal proprietario dell'app rispetto alla configurazione di base mostrata in questa guida.

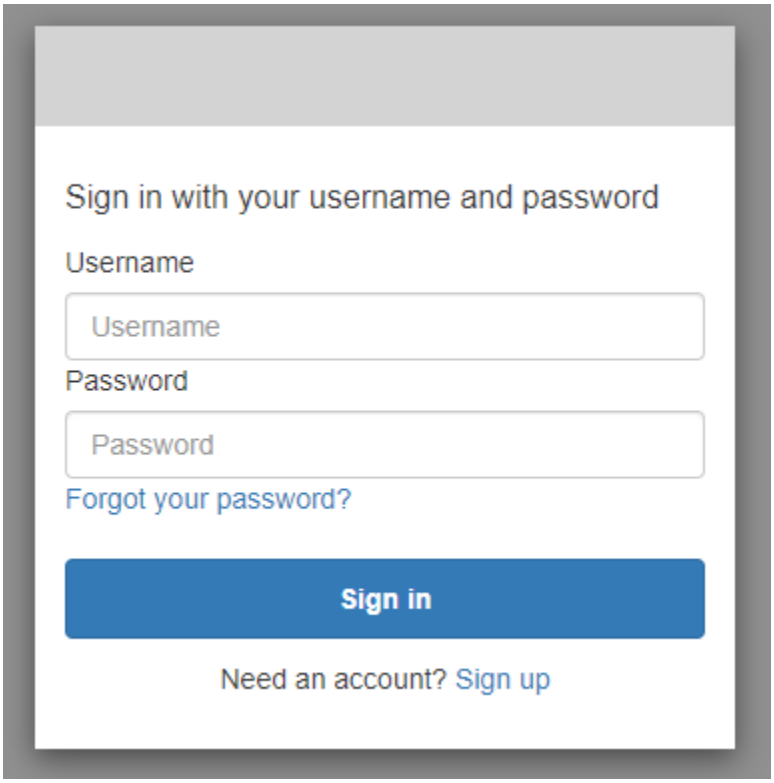
1. A seconda delle opzioni scelte dal proprietario dell'app, potrebbe venire visualizzato un elenco di provider da usare per l'accesso oppure essere necessario immettere un nome utente o una password. Quando si effettua l'accesso immettendo un nome utente e una password in questa pagina, significa che Amazon Cognito è il tuo provider di accesso. In caso contrario, il tuo provider di accesso corrisponde al pulsante che scegli.

   Se normalmente scegli un provider nella pagina di accesso e la password non funziona, segui la procedura per reimpostare la password con il provider. Se Amazon Cognito è il tuo provider di accesso, scegli Forgot your password? (Password dimenticata?).

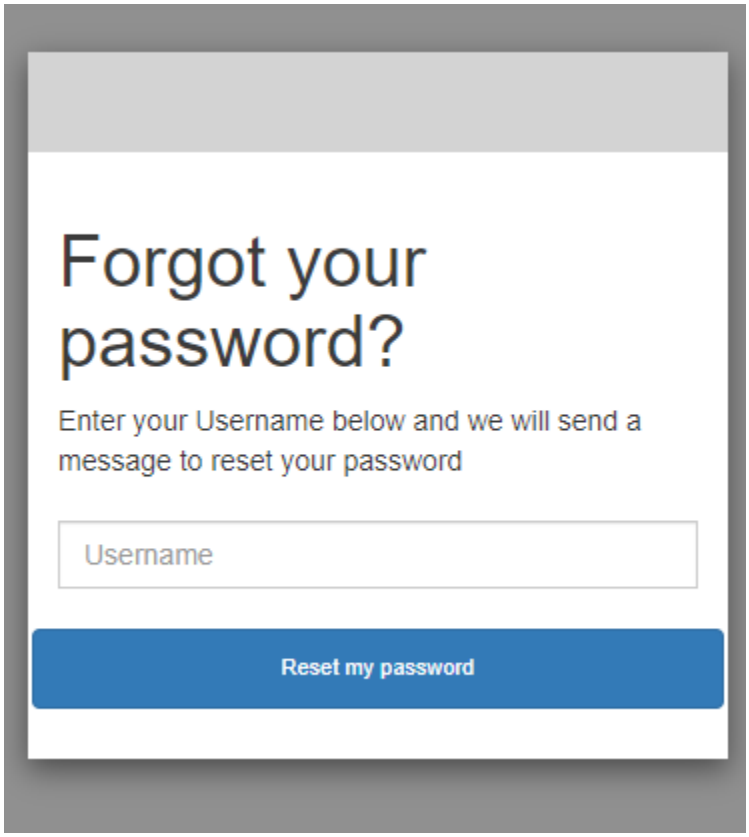
With multiple sign-in providers



With only Amazon Cognito as a sign-in provider

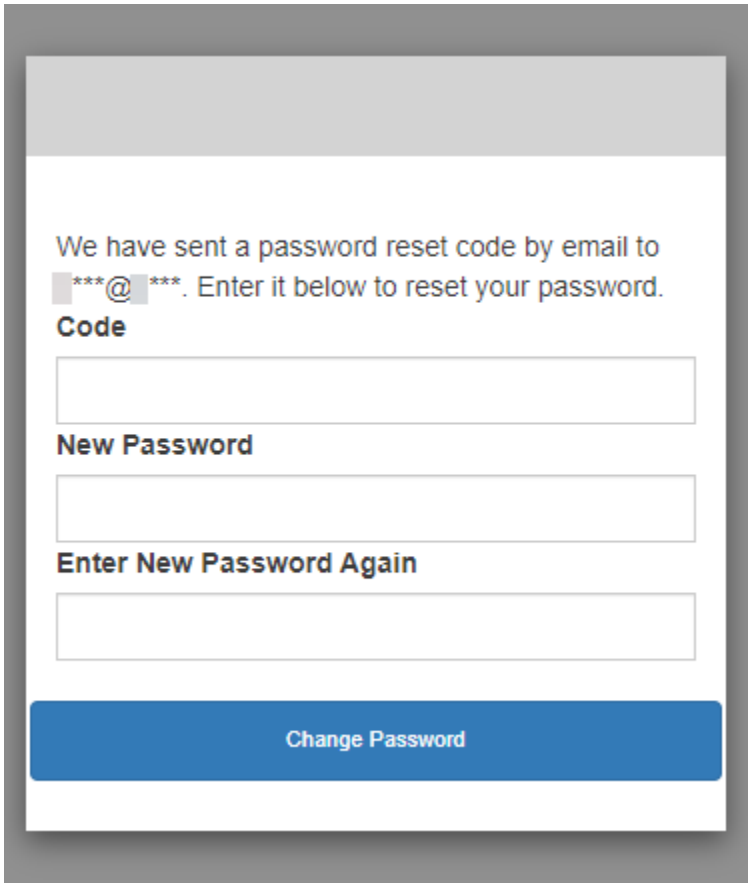


2. Nella pagina Forgot your password? (Password dimenticata?), Amazon Cognito richiede le informazioni utilizzate per eseguire l'accesso. Potrebbe trattarsi del nome utente, dell'indirizzo e-mail o del numero di telefono.



3. Amazon Cognito ti invierà un codice come messaggio e-mail o SMS.

   Inserisci il codice che hai ricevuto e la nuova password due volte nei campi visualizzati. Devi inserire questo codice di ripristino entro 8 minuti.



4. Dopo aver modificato la password, torna alla pagina di accesso e accedi con la nuova password.

Autorizzazione Scopes, M2M e API con server di risorse

Dopo aver configurato un dominio per il pool di utenti, Amazon Cognito effettua automaticamente il provisioning di un server di autorizzazione OAuth 2.0 e di un'interfaccia utente Web ospitata con pagine di registrazione e accesso mostrate dall'app agli utenti. Per ulteriori informazioni, consulta Aggiungi un client di app con l'interfaccia utente ospitata. Puoi scegliere gli ambiti che desideri che il server di autorizzazione aggiunga ai token di accesso. Gli ambiti autorizzano l'accesso ai server di risorse e ai dati degli utenti.

Un server di risorse è un server API OAuth 2.0. Per proteggere le risorse protette dall'accesso, verifica che i token di accesso del pool di utenti contengano gli ambiti che autorizzano il metodo e il percorso richiesti nell'API che protegge. Verifica l'emittente in base alla firma del token, la validità in base alla scadenza del token e il livello di accesso in base agli ambiti delle richieste di token. Gli ambiti del pool di utenti sono indicati nel claim del token di accesso. `scope` Per ulteriori informazioni sulle dichiarazioni nei token di accesso di Amazon Cognito, consulta Utilizzo del token di accesso.

Con Amazon Cognito, gli ambiti dei token di accesso possono autorizzare l'accesso alle API esterne o agli attributi utente. Puoi emettere token di accesso a utenti locali, utenti federati o identità di macchine.

Autorizzazione M achine-to-machine (M2M)

Amazon Cognito supporta applicazioni che accedono ai dati delle API con identità di macchina. Le identità delle macchine nei pool di utenti sono client riservati che vengono eseguiti su server di applicazioni e si connettono a API remote. Il loro funzionamento avviene senza l'interazione dell'utente: attività pianificate, flussi di dati o aggiornamenti delle risorse. Quando questi client autorizzano le proprie richieste con un token di accesso, eseguono l'autorizzazione da macchina a macchina o M2M. Nell'autorizzazione M2M, un segreto condiviso sostituisce le credenziali dell'utente nel controllo degli accessi.

Un'applicazione che accede a un'API con autorizzazione M2M deve avere un ID client e un client secret. Nel tuo pool di utenti, devi creare un client di app che supporti la concessione delle credenziali dei client. Per supportare le credenziali del client, il client dell'app deve disporre di un client secret e devi disporre di un dominio del pool di utenti. In questo flusso, l'identità della macchina richiede un token di accesso direttamente da. Endpoint Token È possibile autorizzare solo ambiti personalizzati dai server di risorse nei token di accesso per la concessione delle credenziali dei client. Per ulteriori informazioni sulla configurazione dei client delle app, consulta. Client dell'app pool di utenti

Il token di accesso derivante dalla concessione delle credenziali di un client è una dichiarazione verificabile delle operazioni che si desidera consentire all'identità della macchina di richiedere a un'API. Per saperne di più su come i token di accesso autorizzano le richieste API, continua a leggere. Per un'applicazione di esempio, consulta Amazon Cognito e l'autorizzazione da macchina a macchina basata su Amazon Cognito e API Gateway tramite AWS CDK.

L'autorizzazione M2M ha un modello di fatturazione diverso dal modo in cui vengono fatturate gli utenti attivi mensili (MAU). Laddove l'autenticazione utente comporta un costo per utente attivo, la fatturazione M2M riflette le credenziali dei clienti attivi, i client delle app e il volume totale delle richieste di token. Per ulteriori informazioni, consultare Prezzi di Amazon Cognito. Per controllare i costi dell'autorizzazione M2M, ottimizza la durata dei token di accesso e il numero di richieste di token effettuate dalle tue applicazioni. Scopri come utilizzare Caching dei token la memorizzazione nella cache di API Gateway per ridurre le richieste di nuovi token nell'autorizzazione M2M.

Per informazioni sull'ottimizzazione delle operazioni di Amazon Cognito che aggiungono costi alla bolletta, AWS consulta. Gestione dei costi

Informazioni sugli ambiti

Un ambito è un livello di accesso che un'app può richiedere a una risorsa. In un token di accesso Amazon Cognito, l'ambito è supportato dal trust che configuri con il tuo pool di utenti: un emittente affidabile di token di accesso con una firma digitale nota. I pool di utenti possono generare token di accesso con scopi che dimostrano che il cliente è autorizzato a gestire alcuni o tutti i propri profili utente o a recuperare dati da un'API di back-end. I pool di utenti di Amazon Cognito emettono token di accesso con l'ambito API riservato del pool di utenti, gli ambiti personalizzati e gli ambiti standard.

L'ambito API riservato dei pool di utenti

L'ambito `aws.cognito.signin.user.admin` autorizza l'API dei pool di utenti Amazon Cognito. Autorizza il titolare di un token di accesso a interrogare e aggiornare tutte le informazioni su un utente del pool di utenti tramite, ad esempio, le operazioni e le GetUserAPI. UpdateUserAttributes Quando autentichi il tuo utente con l'API dei pool di utenti di Amazon Cognito, questo è l'unico ambito che ricevi nel token di accesso. È anche l'unico ambito di cui hai bisogno per leggere e scrivere gli attributi utente che il client dell'app è autorizzato a leggere e scrivere. Puoi anche richiedere questo ambito nelle richieste a Endpoint Authorize. Questo ambito da solo non è sufficiente per richiedere attributi utente da Endpoint UserInfo. Per i token di accesso che autorizzano l'API dei pool di utenti e le richieste `userInfo` per gli utenti, devi richiedere entrambi gli ambiti `openid` e `aws.cognito.signin.user.admin` in una richiesta `/oauth2/authorize`.

Ambiti personalizzati

Gli ambiti personalizzati autorizzano le richieste alle API esterne protette dai server di risorse. Puoi richiedere ambiti personalizzati con altri tipi di ambiti. Puoi trovare maggiori informazioni sugli ambiti personalizzati in questa pagina.

Ambiti standard

Quando autentichi gli utenti con il server di autorizzazione OAuth 2.0 del tuo pool di utenti, inclusa l'interfaccia utente ospitata, devi richiedere gli ambiti. Puoi autenticare gli utenti locali del pool di utenti e gli utenti federati di terze parti nel tuo server di autorizzazione Amazon Cognito. Gli ambiti OAuth 2.0 standard autorizzano l'app a leggere le informazioni utente da Endpoint UserInfo tuo pool di utenti. Il modello OAuth, in cui si interrogano gli attributi utente dall'endpoint `userInfo`, può ottimizzare l'app per un volume elevato di richieste di attributi utente. L'endpoint `userInfo` restituisce gli attributi a un livello di autorizzazione determinato dagli ambiti del token di accesso. Puoi autorizzare il client dell'app a emettere token di accesso con i seguenti ambiti OAuth 2.0 standard.

openid

L'ambito minimo per query OpenID Connect (OIDC). Autorizza il token ID, l'attestazione di identificazione univoca `sub` e la possibilità di richiedere altri ambiti.

> **Note**
>
> Quando richiedi l'ambito `openid` e nessun altro, il token ID del pool di utenti e la risposta `userInfo` includono attestazioni per tutti gli attributi utente che il client dell'app è in grado di leggere. Quando richiedi `openid` e altri ambiti standard come `profile`, `email` e `phone`, il contenuto del token ID e della risposta userInfo è limitato ai vincoli degli ambiti aggiuntivi.
> Ad esempio, una richiesta a Endpoint Authorize con il parametro `scope=openid+email` restituisce un token ID con `sub`, `email` e `email_verified`. Il token di accesso di questa richiesta restituisce gli stessi attributi daEndpoint UserInfo. Una richiesta con parametro `scope=openid` restituisce tutti gli attributi leggibili dal client nel token ID e da `userInfo`.

profilo

Autorizza tutti gli attributi utente che il client dell'app è in grado di leggere.

e-mail

Autorizza gli attributi utente `email` e `email_verified`. Amazon Cognito restituisce `email_verified` se un valore è stato impostato in modo esplicito.

telefono

Autorizza gli attributi utente `phone_number` e `phone_number_verified`.

Informazioni sui server di risorse

Un'API del server di risorse potrebbe consentire l'accesso alle informazioni in un database o controllare le risorse IT. Un token di accesso di Amazon Cognito può autorizzare l'accesso alle API che supportano OAuth 2.0. Le REST API Gateway Amazon API sono caratterizzate dal supporto integrato per l'autorizzazione con i token di accesso di Amazon Cognito. L'app trasferisce il token d'accesso nella chiamata API ai server di risorse. Il server di risorse ispeziona il token d'accesso per determinare se gli accessi devono essere concessi.

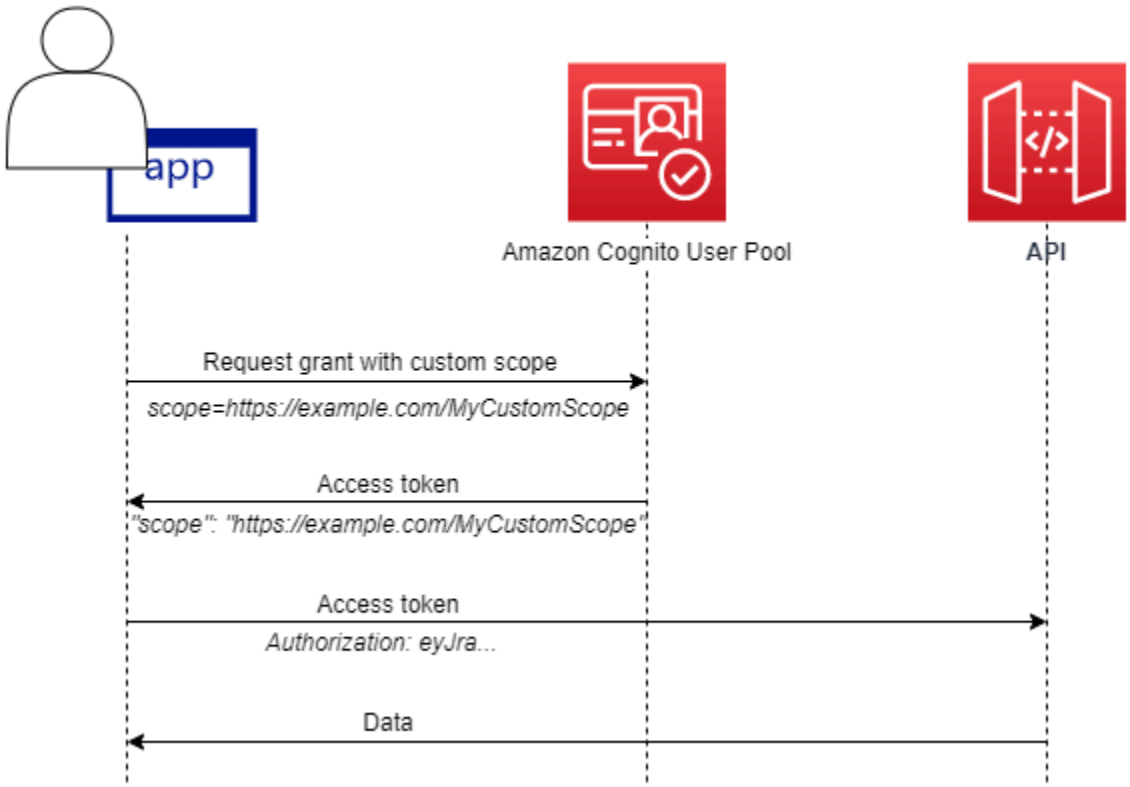
Amazon Cognito potrebbe apportare future modifiche allo schema dei token di accesso del pool di utenti. Se la tua app analizza il contenuto del token di accesso prima di passarlo a un'API, devi progettare il codice per accettare gli aggiornamenti dello schema.

Gli ambiti personalizzati vengono definiti dall'utente ed estendono le funzionalità di autorizzazione di un pool di utenti per includere scopi non correlati all'interrogazione e alla modifica degli utenti e dei relativi attributi. Ad esempio, se hai un server di risorse per l'archiviazione di foto, questo potrebbe definire due ambiti, `photos.read` per l'accesso in lettura alle foto e `photos.write` per l'accesso in scrittura/eliminazione. Puoi configurare un'API per accettare i token di accesso per l'autorizzazione e la concessione di richieste `HTTP GET` per accedere ai token con `photos.read` nell'attestazione `scope` e richieste `HTTP POST` ai token con `photos.write`. Questi sono ambiti personalizzati.

> **Note**
>
> Per elaborare una qualsiasi richiesta all'interno del token, il tuo server di risorse deve verificare la firma e la data di scadenza dei token d'accesso. Per ulteriori informazioni sulla verifica dei token, consulta Verifica di un JSON Web Token. Per ulteriori informazioni sulla verifica e sull'utilizzo dei token del pool di utenti in Gateway Amazon API, consulta la sezione Integrazione dei pool di utenti Amazon Cognito con API Gateway. API Gateway è una buona soluzione per l'ispezione dei token d'accesso e la protezione delle risorse. Per ulteriori informazioni sulle autorizzazioni Lambda di API Gateway, consulta Uso di autorizzazioni Lambda di API Gateway.

Panoramica

Con Amazon Cognito, puoi creare server di risorse OAuth2.0 e associarvi ambiti personalizzati. Gli ambiti personalizzati in un token di accesso autorizzano azioni specifiche nell'API. Puoi autorizzare qualsiasi app del client nel pool di utenti a emettere ambiti personalizzati da uno qualsiasi dei server di risorse. Associa ambiti personalizzati a un client dell'app e richiedi questi ambiti nelle concessioni del codice di autorizzazione OAuth2.0, nelle concessioni implicite e nelle concessioni delle credenziali client dal Endpoint Token. Amazon Cognito aggiunge ambiti personalizzati nell'attestazione `scope` in un token di accesso. Un client può utilizzare il token d'accesso piuttosto che il suo server di risorse, facendo in modo che le decisioni legate all'autorizzazione siano basate sugli ambiti presenti nel token. Per ulteriori informazioni sui token d'accesso proposto, consulta Utilizzo dei token con i bacini d'utenza.



Per ottenere un token di accesso con ambiti personalizzati, l'app deve effettuare una richiesta all'Endpoint Token per riscattare un codice di autorizzazione o per richiedere la concessione delle credenziali del client. Nell'interfaccia utente ospitata, puoi anche richiedere ambiti personalizzati in un token di accesso da una concessione implicita.

> **Note**
>
> Perché sono progettate per l'autenticazione interattiva con il pool di utenti come IdP InitiateAuthe AdminInitiateAuthle richieste producono solo un'`scope`attestazione nel token di accesso con il singolo valore. `aws.cognito.signin.user.admin`

Gestione del server di risorse e degli ambiti personalizzati

Durante la creazione di un server di risorse, è necessario fornire un nome per il server di risorse e un identificatore per il server di risorse. Per ogni ambito che crei nel server di risorse, è necessario fornire nome e descrizione.

- Nome del server di risorse: un nome intuitivo per il server di risorse, ad esempio `Solar system object tracker` o `Photo API`.
- Identificatore del server di risorse: un identificatore univoco per il server di risorse. L'identificatore è qualsiasi nome che desideri associare alla tua API, ad esempio `solar-system-data`. Puoi configurare identificatori più lunghi, ad esempio `https://solar-system-data-api.example.com` come riferimento più diretto ai percorsi URI delle API, ma stringhe più lunghe aumentano la dimensione dei token di accesso.
- Nome ambito: il valore che desideri nell'attestazione `scope`. Ad esempio, `sunproximity.read`.
- Descrizione: una descrizione intuitiva dell'ambito. Ad esempio, `Check current proximity to sun`.

Amazon Cognito può includere ambiti personalizzati nei token di accesso per qualsiasi utente, locale nel pool di utenti o federato, con un provider di identità di terze parti. Puoi scegliere gli ambiti per i token di accesso degli utenti durante i flussi di autenticazione con il server di autorizzazione OAuth 2.0 che include l'interfaccia utente ospitata. L'autenticazione dell'utente deve iniziare in corrispondenza di Endpoint Authorize con `scope` come uno dei parametri della richiesta. Di seguito è riportato un formato consigliato per i server di risorse. Per un identificatore, usa un nome intuitivo per le API. Per un ambito personalizzato, usa l'azione che autorizzano.

```
resourceServerIdentifier/scopeName
```

Ad esempio, hai scoperto un nuovo asteroide nella fascia di Kuiper e desideri registrarlo tramite la tua API `solar-system-data`. L'ambito che autorizza le operazioni di scrittura nel database degli asteroidi è `asteroids.add`. Quando richiedi il token di accesso che ti autorizzerà a registrare la tua scoperta, formatta il parametro di richiesta HTTPS `scope` come `scope=solar-system-data/asteroids.add`.

L'eliminazione di un ambito dal server di risorse non elimina la relativa associazione con tutti i client. Invece, l'ambito è contrassegnato come inattivo. Amazon Cognito non aggiunge ambiti inattivi ai token di accesso, ma per il resto procede normalmente se l'app ne richiede uno. Se aggiungi nuovamente l'ambito al tuo server di risorse in un secondo momento, Amazon Cognito lo scrive nuovamente nel token di accesso. Se richiedi un ambito che non hai associato all'app del client, a prescindere che venga eliminato dal server di risorse del pool di utenti, l'autenticazione non va a buon fine.

Puoi utilizzare l'API o AWS Management Console la CLI per definire server di risorse e ambiti per il tuo pool di utenti.

Definizione di un server di risorse per il tuo bacino d'utenza (AWS Management Console)

È possibile utilizzare il AWS Management Console per definire un server di risorse per il pool di utenti.

Definizione di un server di risorse

1. Accedi alla console Amazon Cognito.
2. Nel riquadro di navigazione, seleziona User Pools (bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Seleziona la scheda App integration (integrazione app) e individua l'opzione Resource servers (server di risorse).
4. Seleziona Create a resource server (Crea un server di risorse).
5. Inserisci un nome del server di risorse. Ad esempio, `Photo Server`.
6. Inserisci un Resource server identifier (identificatore del server di risorse). Ad esempio, `com.example.photos`.
7. Inserisci gli Custom scopes (ambiti personalizzati) per le risorse, come `read` e `write`.
8. Per ogni Scope name (nome ambito), inserisci una Description (descrizione), come `view your photos` e `update your photos`.
9. Scegli Crea.

I tuoi ambiti personalizzati possono essere esaminati nella scheda Integrazione app alla voce Resource servers (Server di risorse), nella colonna Custom scopes (ambiti personalizzati). Gli ambiti personalizzati possono essere abilitati per i client app dalla scheda Integrazione app (App integration) alla voce App clients (client dell'App). Seleziona un client dell'App, individua l'opzione Hosted UI settings (impostazioni dell'interfaccia utente ospitata) e seleziona Edit (modifica). Aggiungi degli Custom scopes (ambiti personalizzati) e seleziona Save changes (salva modifiche).

Definizione di un server di risorse per il pool di utenti (AWS CLI e AWS l'API)

Usa i seguenti comandi per specificare le impostazioni dei server di risorse per il tuo bacino d'utenza.

Creazione di un server di risorse

- AWS CLI: `aws cognito-idp create-resource-server`
- AWS API: CreateResourceServer

Ottenimento di informazioni sulle impostazioni dei tuoi server di risorse

- AWS CLI: `aws cognito-idp describe-resource-server`
- AWS API: DescribeResourceServer

Creazione di liste di informazioni su tutti i server di risorse per il tuo bacino d'utenza

- AWS CLI: `aws cognito-idp list-resource-servers`
- AWS API: ListResourceServers

Eliminazione di un server di risorse

- AWS CLI: `aws cognito-idp delete-resource-server`
- AWS API: DeleteResourceServer

Aggiornamento delle impostazioni per un server di risorse

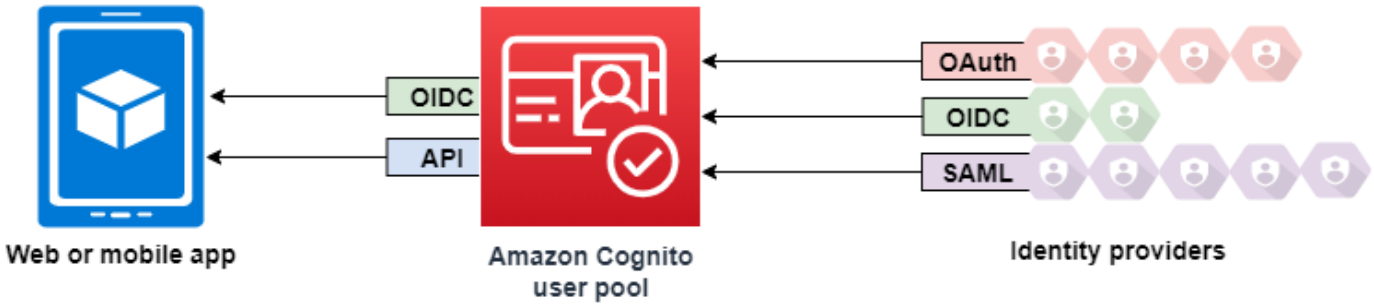
- AWS CLI: `aws cognito-idp update-resource-server`
- AWS API: UpdateResourceServer

Aggiunta di un accesso al bacino d'utenza tramite terze parti

Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Con l'interfaccia utente web ospitata integrata, Amazon Cognito fornisce la gestione e la gestione dei token per tutti gli utenti autenticati. IdPs In questo modo, i sistemi back-end possono standardizzare un set di token del bacino d'utenza.

Funzionamento dell'accesso federato nei pool di utenti di Amazon Cognito

L'accesso tramite terze parti (federazione) è disponibile per i bacini d'utenza Amazon Cognito. Questa funzione è indipendente dalla federazione tramite pool di identità di Amazon Cognito (identità federate).



Amazon Cognito è una directory utente e un gestore dell'identità digitale (IdP) OAuth 2.0. Quando gli utenti locali effettuano l'accesso alla directory Amazon Cognito, il pool di utenti è un IdP dell'app. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno.

Quando connetti Amazon Cognito a social, SAML o OpenID Connect (OIDC IdPs), il tuo pool di utenti funge da ponte tra più fornitori di servizi e la tua app. Per il gestore dell'identità digitale (IdP) in uso, Amazon Cognito è un fornitore di servizi. Devi IdPs passare un token ID OIDC o un'asserzione SAML ad Amazon Cognito. Amazon Cognito legge le asserzioni relative all'utente nel token o nell'asserzione e mappa tali asserzioni su un nuovo profilo utente nella directory del pool di utenti.

Amazon Cognito crea quindi un profilo utente per l'utente federato nella propria directory. Amazon Cognito aggiunge attributi all'utente in base alle asserzioni del gestore dell'identità digitale (IdP) e, nel caso di gestori dell'identità OIDC e digitale, un endpoint `userinfo` pubblico gestito dal gestore dell'identità digitale. Gli attributi dell'utente cambiano nel pool di utenti quando un attributo del gestore dell'identità digitale (IdP) mappato cambia. Puoi anche aggiungere altri attributi indipendentemente da quelli del gestore dell'identità digitale (IdP).

Dopo che Amazon Cognito ha creato un profilo per l' utente federato, cambia la sua funzione e si presenta come gestore dell'identità digitale (IdP) all'app, che ora è il fornitore di servizi. Amazon Cognito è una combinazione di gestori dell'identità digitale OIDC e OAuth 2.0. Genera token di accesso, token ID e token di aggiornamento. Per ulteriori informazioni sui token, consulta Utilizzo di token con bacini d'utenza.

Devi progettare un'app che si integri con Amazon Cognito per autenticare e autorizzare gli utenti, che siano federati o nativi.

Responsabilità di un'app come provider di servizi con Amazon Cognito

Verifica ed elaborazione delle informazioni nei token

Nella maggior parte degli scenari, Amazon Cognito reindirizza l'utente autenticato a un URL dell'app aggiunta con un codice di autorizzazione. L'app scambia il codice per l'accesso, l'ID e l'aggiornamento dei token. Deve quindi verificare la validità dei token e inviare informazioni all'utente in base alle asserzioni contenute nei token.

Risposta agli eventi di autenticazione con le richieste API di Amazon Cognito

L'app deve integrarsi con le API dei pool di utenti di Amazon Cognito e gliendpoint dell'API di autenticazione. L'API di autenticazione effettua l'accesso e la disconnessione dell'utente e gestisce i token. L'API dei pool di utenti dispone di una serie di operazioni che gestiscono il pool di utenti, gli utenti e la sicurezza dell'ambiente di autenticazione. L'app deve sapere cosa fare dopo aver ricevuto una risposta da Amazon Cognito.

Informazioni importanti sull'accesso di terze parti ai pool di utenti di Amazon Cognito

- Se i tuoi utenti accedono con provider federati, devi scegliere un dominio. Configura l'interfaccia utente ospitata di Amazon Cognito e gli endpoint OIDC e dell'interfaccia utente ospitati. Per ulteriori informazioni, consulta Utilizzo del proprio dominio per l'interfaccia utente ospitata.
- Non puoi accedere a utenti federati con operazioni API come e. InitiateAuthAdminInitiateAuth Gli utenti federati possono accedere solo con l'Endpoint Login o l'Endpoint Authorize.
- Endpoint Authorize è un endpoint di reindirizzamento. Se fornisci un parametro `idp_identifier` o `identity_provider` nella tua richiesta, viene reindirizzato automaticamente al tuo IdP, ignorando l'interfaccia utente ospitata. In caso contrario, viene reindirizzato all'interfaccia utente ospitata Endpoint Login. Per vedere un esempio, consulta Scenario di esempio: aggiungi ai preferiti le app Amazon Cognito in una dashboard aziendale.
- Quando l'interfaccia utente ospitata reindirizza una sessione a un provider di identità federato, Amazon Cognito include l'intestazione `user-agent Amazon/Cognito` nella richiesta.
- Amazon Cognito deriva l'attributo `username` per un profilo utente federato da una combinazione di un identificatore fisso e il nome del gestore dell'identità digitale (IdP). Per generare un nome utente che soddisfi i requisiti personalizzati, crea una mappatura per l'attributo `preferred_username`. Per ulteriori informazioni, consulta Cose da sapere sulle mappature.

Esempio: `MyIDP_bob@example.com`

- Amazon Cognito registra le informazioni sull'identità dell'utente federato in un attributo e un'asserzione nel token ID, chiamato `identities`. Questa asserzione contiene il gestore dell'utente e il relativo ID univoco del gestore. Non è possibile modificare l'attributo `identities` direttamente in un profilo utente. Per ulteriori informazioni su come collegare un utente federato, consulta Collegamento di utenti federati a un profilo utente esistente.
- Quando aggiorni il tuo IdP in una richiesta API UpdateIdentityProvider, la visualizzazione delle modifiche nell'interfaccia utente ospitata può richiedere fino a un minuto.
- Amazon Cognito supporta fino a 20 reindirizzamenti HTTP tra se stesso e l'IdP.
- Quando l'utente effettua l'accesso con l'interfaccia utente ospitata, il relativo browser memorizza un cookie di sessione di accesso crittografato che registra il client e il provider con cui è stato effettuato l'accesso. Se tenta di accedere nuovamente con gli stessi parametri, l'interfaccia utente ospitata riutilizza l'eventuale sessione esistente non scaduta e l'utente si autentica senza fornire nuovamente le credenziali. Se l'utente accede nuovamente con un IdP diverso, incluso un passaggio da o verso un accesso al pool di utenti locale, deve fornire le credenziali e generare una nuova sessione di accesso.

  Puoi assegnare qualsiasi gruppo di utenti IdPs a qualsiasi client dell'app e gli utenti possono accedere solo con un IdP che hai assegnato al loro client di app.

Argomenti

- Configurazione dei provider di identità per il bacino d'utenza
- Utilizzo di provider di identità social con un pool di utenti
- Utilizzo di provider di identità SAML con un pool di utenti
- Utilizzo di provider di identità OIDC con un pool di utenti
- Specificazione di mappature degli attributi del provider di identità per il bacino d'utenza
- Collegamento di utenti federati a un profilo utente esistente

Configurazione dei provider di identità per il bacino d'utenza

Nella scheda Esperienza di accesso sotto Accesso con provider di identità federato, puoi aggiungere provider di identità (IdPs) al tuo pool di utenti. Per ulteriori informazioni, consulta Aggiunta di un accesso al bacino d'utenza tramite terze parti.

Argomenti

- Configurazione dell'accesso dell'utente con un IdP social
- Configurazione dell'accesso utente con un IdP OIDC
- Configurazione dell'accesso utente con un IdP SAML

Configurazione dell'accesso dell'utente con un IdP social

Puoi utilizzare la federazione per i bacini d'utenza di Amazon Cognito per integrare i provider di identità social, come Facebook, Google e Login with Amazon.

Per aggiungere un provider di identità social, per prima cosa, crea un account per sviluppatori con il provider di identità. Dopo aver creato un account per sviluppatori, registra l'app con il provider di identità. Il provider di identità crea un'ID e un segreto per l'app, i cui valori devono essere configurati nei bacini d'utenza di Amazon Cognito.

- Piattaforma di identità Google
- Facebook per sviluppatori
- Login with Amazon
- Accedi con Apple

Per integrare l'accesso utente con un IdP social

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Seleziona la scheda Sign-in experience (Esperienza di accesso) e individua Federated sign-in (Accesso federato).
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure seleziona il provider di identità Facebook, Google, Amazon o Apple che hai configurato, individua la voce Identity provider information (Informazioni provider di identità) e seleziona Edit (Modifica). Per ulteriori informazioni su come aggiungere provider di identità social consulta Utilizzo di provider di identità social con un pool di utenti.
5. Inserisci le informazioni del tuo provider di identità social completando uno dei seguenti passaggi, in base alla tua scelta di IdP:

Per Facebook, Google e Login with Amazon:

Inserisci l'ID dell'app e il segreto app ricevuti al momento della creazione dell'app client.

Accedi con Apple

Inserisci l'ID del servizio fornito ad Apple, nonché l'ID del team, l'ID della chiave e la chiave privata ricevuti quando è stata creato il client dell'app.

6. Nel campo Authorized scopes (Ambiti autorizzati), inserisci i nomi degli ambiti dei provider di identità social che intendi mappare agli attributi del bacino d'utenza. Gli ambiti definiscono gli attributi utente, ad esempio nome ed indirizzo e-mail con cui intendi accedere con l'App. Quando inserisci gli ambiti, usa le seguenti linee guida in base alla tua scelta di IdP:

   - Facebook: ambiti separati da virgole. Per esempio:

     ```
     public_profile, email
     ```

   - Google, Login with Amazon e Accedi con Apple: ambiti separati da spazi. Per esempio:
     - Google: `profile email openid`
     - Login with Amazon: `profile postal_code`
     - Accedi con Apple: `name email`

     > **Note**
     >
     > Per il servizio Accedi con Apple (console), utilizza le caselle di controllo per selezionare gli ambiti.

7. Scegli Save changes (Salva modifiche).
8. Dalla scheda App client integration (Integrazione del client dell'app), seleziona uno degli App clients (Client dell'app) nell'elenco e quindi seleziona l'opzione Edit hosted UI settings (Modifica impostazioni interfaccia utente ospitata). Aggiungi il nuovo provider di identità social al client dell'App alla voce Identity providers (Provider di identità).
9. Scegli Save changes (Salva modifiche).

Per ulteriori informazioni sui social IdPs, consulta. Utilizzo di provider di identità social con un pool di utenti

Configurazione dell'accesso utente con un IdP OIDC

Puoi integrare l'accesso degli utenti tramite un provider di identità OpenID Connect (OIDC), ad esempio Salesforce o Ping Identity.

Per aggiungere un provider OIDC a un pool di utenti

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Scegli User Pools (Bacini d'utenza) dal menu di navigazione.
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua l'opzione Federated sign-in (Accesso federato) e seleziona Add an identity provider (Aggiungi un provider di identità).
5. Scegli un provider di identità OpenID Connect.
6. Inserisci un nome univoco nel campo Provider name (Nome provider).
7. Inserisci l'ID client che hai ricevuto dal tuo provider nel campo Client ID (ID client).
8. Inserisci il segreto client che hai ricevuto dal tuo provider nel campo Client secret (Segreto client).
9. Inserisci gli Authorized scopes (Ambiti autorizzati) per questo provider. Gli ambiti definiscono quali gruppi di attributi utente (ad esempio `name` e `email`) verranno richiesti dalla tua applicazione al tuo provider. Gli ambiti devono essere separati da spazi, secondo la specifica OAuth 2.0.

   All'utente viene richiesto il consenso a fornire questi attributi alla tua applicazione.
10. Scegli un Attribute request method (Metodo della richiesta di attributo) per fornire ad Amazon Cognito il metodo HTTP (GET o POST) da utilizzare per recuperare i dettagli dell'utente dall'endpoint userInfo gestito dal tuo provider.
11. Scegli un Setup method (Metodo di impostazione) per recuperare gli endpoint OpenID Connect dall'opzione Auto fill through issuer URL (Riempimento automatico attraverso URL dell'emittente) o da Manual input (Inserimento manuale). Utilizza lo strumento Auto fill through issuer URL (Riempimento automatico dell'URL dell'emittente) quando il tuo provider ha un endpoint pubblico `.well-known/openid-configuration` in cui Amazon Cognito può recuperare gli URL degli endpoint `authorization`, `token`, `userInfo` e `jwks_uri`.
12. Inserisci gli URL dell'emittente o degli endpoint `authorization`, `token`, `userInfo`, e `jwks_uri` dal provider di IdP.

> **Note**
>
> È possibile utilizzare solo i numeri di porta 443 e 80 con URL di rilevamento, compilati automaticamente e inseriti manualmente. Gli accessi utente non riescono se il provider OIDC utilizza porte TCP non standard.
> L'URL dell'emittente deve iniziare con `https://` e non deve terminare con un carattere `/`. Ad esempio, Salesforce usa questo URL: `https://login.salesforce.com`
> Il documento `openid-configuration` associato all'URL dell'emittente deve fornire URL HTTPS per i seguenti valori:`authorization_endpoint`, `token_endpoint`, `userinfo_endpoint` e `jwks_uri`. Allo stesso modo, quando scegli Manual input (Inserimento manuale), è possibile inserire solo URL HTTPS.

13. Per impostazione predefinita, la richiesta OIDC sub viene mappata all'attributo del bacino d'utenza Username. Puoi mappare altre richieste OIDC agli attributi del bacino d'utenza. Inserisci la richiesta OIDC e seleziona l'attributo del bacino d'utenza corrispondente dall'elenco a discesa. Ad esempio, l'indirizzo e-mail della richiesta viene spesso mappato all'attributo del bacino d'utenza Email (E-mail).
14. Mappa gli attributi aggiuntivi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, consulta la sezione Specificazione di mappature degli attributi del provider di identità per il bacino d'utenza.
15. Scegli Create (Crea).
16. Dalla scheda App client integration (Integrazione client dell'app), seleziona uno dei client dell'app nella lista e quindi Edit hosted UI settings (Modifica le impostazioni dell'interfaccia utente ospitata). Aggiungi il nuovo provider di identità OIDC al client dell'app alla voce Identity providers (Provider di identità).
17. Scegli Save changes (Salva modifiche).

Per ulteriori informazioni su OIDC IdPs, vedere. Utilizzo di provider di identità OIDC con un pool di utenti

Configurazione dell'accesso utente con un IdP SAML

Puoi utilizzare la federazione per il bacino d'utenza di Amazon Cognito per l'integrazione con un provider di identità (IdP) SAML. Puoi fornire un documento di metadati, o caricando il file oppure

inserendo un URL di endpoint del documento di metadati. Per informazioni su come ottenere documenti di metadati per IdPs SAML di terze parti, consulta. Configurazione del provider di identità SAML di terze parti

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua l'opzione Federated sign-in (Accesso federato) e seleziona Add an identity provider (Aggiungi un provider di identità).
5. Scegli un provider di identità SAML.
6. Inserisci gli Identifiers (Identificatori) separati da virgole. Un identificatore indirizza Amazon Cognito a controllare l'indirizzo e-mail di accesso dell'utente e poi indirizza l'utente al provider che corrisponde al suo dominio..
7. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. Configura il provider di identità SAML 2.0 per inviare le risposte di disconnessione all'endpoint `https://mydomain.us-east-1.amazoncognito.com/saml2/logout` creato da Amazon Cognito quando configuri l'interfaccia utente ospitata. L'endpoint `saml2/logout` utilizza POST vincolanti.

   > **Note**
   >
   > Se selezioni questa opzione e il provider di identità SAML si aspetta una richiesta di disconnessione con firma, è necessario configurare anche il certificato di firma fornito da Amazon Cognito con il tuo IdP SAML.
   > L'IdP SAML elabora la richiesta di disconnessione con firmata e disconnette l'utente dalla sessione Amazon Cognito.

8. Seleziona una Metadata document source (Fonte del documento di metadati). Se il tuo provider di identità fornisce metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

> **Note**
>
> Se il provider ha un endpoint pubblico, suggeriamo di fornire l'URL di un documento di metadati anziché caricare un file. Se utilizzi l'URL, Amazon Cognito aggiorna automaticamente i metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

9. Scegli l'opzione Map attributes between your SAML provider and your app (Mappa gli attributi tra il provider SAML e la tua app) per mappare gli attributi del provider SAML al profilo utente nel bacino d'utenza. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

   Ad esempio, quando seleziona l'`email` dell'Attributo del bacino d'utenza, inserisci il nome dell'attributo SAML così come compare nell'asserzione SAML dal provider di identità. Il tuo provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni provider di identità utilizzano nomi semplici, come `email`, mentre altri utilizzano attributi con formato URL simili a questo:

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

10. Scegli Crea.

> **Note**
>
> Se visualizzi `InvalidParameterException` durante la creazione di un provider di identità SAML con un URL di endpoint di metadati HTTPS, assicurati che l'endpoint di metadati abbia l'SSL configurato correttamente e che vi sia associato un certificato SSL valido. Un esempio di tale eccezione è "Errore nel recupero dei metadati da*<metadata endpoint>*" ("Errore nel recupero dei metadati da <metadata endpoint>").

Per configurare l'IdP SAML per l'aggiunta di un certificato di firma

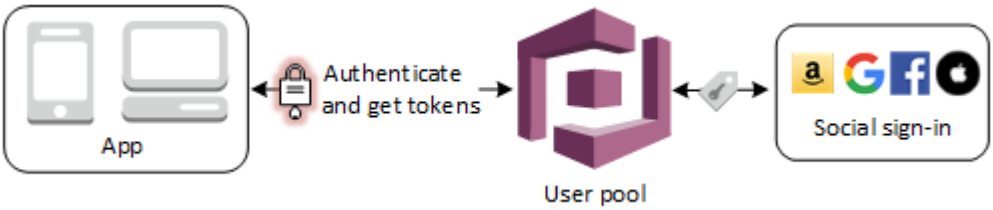
- Per ottenere il certificato contenente la chiave pubblica utilizzata dall'IdP per verificare la richiesta di disconnessione con firma, scegli Mostra certificato di firma in Attiva provider SAML nella finestra di dialogo SAML in Provider di identità nella pagina della console Federazione.

Per ulteriori informazioni su IdPs SAML, consulta. Utilizzo di provider di identità SAML con un pool di utenti

Utilizzo di provider di identità social con un pool di utenti

Gli utenti di app Web e per dispositivi mobili possono accedere tramite provider di identità social (IdP) come Facebook, Google, Amazon e Apple. Con l'interfaccia utente Web ospitata integrata, Amazon Cognito fornisce la gestione dei token per gli utenti autenticati. In questo modo, i sistemi back-end possono standardizzare un set di token del bacino d'utenza. È necessario abilitare l'interfaccia utente ospitata per l'integrazione con i provider di identità social supportati. Quando Amazon Cognito crea la tua interfaccia utente ospitata, crea endpoint OAuth 2.0 che Amazon Cognito, OIDC e social utilizzano per lo scambio di informazioni. IdPs Per ulteriori informazioni, consulta la Documentazione di riferimento delle API Auth dei bacini d'utenza di Amazon Cognito.

Puoi aggiungere un IdP social in oppure puoi utilizzare AWS Management Console la AWS CLI o l'API Amazon Cognito.



> **Note**
>
> L'accesso tramite terze parti (federazione) è disponibile per i bacini d'utenza Amazon Cognito. Questa funzione è indipendente dalla federazione tramite pool di identità di Amazon Cognito (identità federate).

Argomenti

- Prerequisiti
- Fase 1: registrazione con un IdP social
- Fase 2: aggiunta di un IdP social al bacino d'utenza
- Fase 3: test della configurazione dell'IdP social

Prerequisiti

Prima di iniziare, è necessario:

- Un bacino d'utenza con un client di applicazioni e dominio per il bacino d'utenza. Per ulteriori informazioni, consulta Creazione di un bacino d'utenza.
- Un provider di identità social.

Fase 1: registrazione con un IdP social

Prima di creare un IdP social con Amazon Cognito, è necessario registrare l'applicazione con l'IdP social in modo da ricevere un ID client e un segreto client.

Registrazione di un'app con Facebook

1. Creazione di un account sviluppatore con Facebook.
2. Accedi con le tue credenziali di Facebook.
3. Nel menu My Apps (Le mie app), scegli Create New App (Crea nuova app).
4. Inserisci un nome per l'app Facebook, quindi scegli Create App ID (Crea ID dell'app).
5. Nella barra di navigazione a sinistra, scegli Settings (Impostazioni) e successivamente Basic (Di base).
6. Prendi nota di ID app e Segreto app. Li utilizzerai nella sezione successiva.
7. Nella parte inferiore della pagina, scegli +Add Platform (+Aggiungi piattaforma).
8. Scegli Website (Sito Web).
9. Alla voce Website (Sito Web), inserisci il percorso della pagina di accesso per l'app in URL del sito.

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

10. Scegli Save changes (Salva modifiche).
11. Inserisci il percorso della directory principale del dominio del bacino d'utenza in App Domains (Domini app).

```
https://mydomain.us-east-1.amazoncognito.com
```

12. Scegli Save changes (Salva modifiche).
13. Nella barra di navigazione, scegli Add Products (Aggiungi prodotti) e successivamente Set up (Imposta) per il prodotto Facebook Login (Accedi con Facebook).
14. Nella barra di navigazione, scegli Facebook Login (Accedi con Facebook) e successivamente Settings (Impostazioni).

    Inserisci il percorso dell'endpoint `/oauth2/idpresponse` per il dominio del pool di utenti in Valid OAuth Redirect URIs (URI di reindirizzamento OAuth validi).

    ```
    https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
    ```

15. Scegli Save changes (Salva modifiche).

Registrazione di un'app con Amazon

1. Creazione di un account sviluppatore con Amazon.
2. Accedi con le tue credenziali di Amazon.
3. È necessario creare un profilo di sicurezza Amazon per ricevere l'ID client di Amazon e il segreto client.

   Nella barra di navigazione disponibile nella parte superiore della pagina, scegli Apps and Services (App e servizi) e successivamente Login with Amazon (Accedi con Amazon).
4. Scegli Create a Security Profile (Crea un profilo di sicurezza).
5. Digita un Security Profile Name (nome profilo sicurezza), una Security Profile Description (descrizione profilo sicurezza) e un Consent Privacy Notice URL (URL consenso informativa privacy).
6. Scegli Save (Salva).
7. Scegli Client ID (ID client) e Client Secret (Segreto client) per mostrare i dati relativi. Li utilizzerai nella sezione successiva.
8. Passa il mouse sull'icona a forma di ingranaggio e scegli Web Settings (Impostazioni Web), quindi Scegli Edit (Modifica).
9. Inserisci il dominio del bacino d'utenza nel campo Allowed Origins (origini consentite).

   ```
   https://mydomain.us-east-1.amazoncognito.com
   ```

10. Digita il dominio del bacino d'utenza con l'endpoint `/oauth2/idpresponse` in Allowed Return URLs (URL restituiti consentiti).

    ```
    https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
    ```

11. Scegli Save Salva.

Registrazione di un'app con Google

Per ulteriori informazioni su OAuth 2.0 nella piattaforma Google Cloud, consulta la sezione relativa all'autenticazione e autorizzazione nella documentazione disponibile nella pagina Google Workspace for Developers.

1. Creazione di un account sviluppatore con Google.
2. Accedi alla console Piattaforma Google Cloud.
3. Nella barra di navigazione in alto, scegli Select a project (Seleziona un progetto). Se hai già un progetto nella piattaforma Google, nel menu viene visualizzato il tuo progetto predefinito.
4. Scegli Nuovo progetto.
5. Immetti un nome per il progetto e scegli Crea.
6. Nella barra di navigazione a sinistra, scegli APIs and Services (API e servizi), quindiOauth consent screen (Schermata consenso Oauth).
7. Inserisci le informazioni sull'app, un App domain (Dominio dell'app), gli Authorized domains (Domini autorizzati) e le Developer contact information (Informazioni di contatto dello sviluppatore). Gli Authorized domains (Domini autorizzati) devono includere `amazoncognito.com` e la radice del dominio personalizzato, ad esempio `example.com`. Seleziona Salva e continua.
8. 1. In Scopes (Ambiti), scegli Add or remove scopes (Aggiunta o rimozione di ambiti) e seleziona almeno i seguenti ambiti OAuth.

    1. `.../auth/userinfo.email`
    2. `.../auth/userinfo.profile`
    3. openid
9. In Test users (Utenti di test), scegli Add users (Aggiungi utenti). Inserisci l'indirizzo e-mail e qualsiasi altro utente di test autorizzato, quindi scegli SAVE AND CONTINUE (Salva e continua).
10. Espandi di nuovo la barra di navigazione a sinistra e seleziona APIs and Services (API e servizi), quindi Credentials (Credenziali).

11. Scegli CREATE CREDENTIALS (Crea credenziali), quindi OAuth client ID (ID client OAuth).
12. Scegli un Application type (Tipo di applicazione) e assegna un Name (Nome) al client.
13. In JavaScript Origini autorizzate, scegli AGGIUNGI URI. Immetti il dominio del pool di utenti.

    ```
    https://mydomain.us-east-1.amazoncognito.com
    ```

14. In Authorized redirect URIs (URI di reindirizzamento autorizzati), scegli ADD URI (Aggiungi URI). Inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

    ```
    https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
    ```

15. Scegli CREATE (Crea).
16. Memorizza in modo sicuro i valori visualizzati da Google in Your client ID (Il tuo ID client) e Your client secret (Il tuo segreto del client). Fornisci questi valori ad Amazon Cognito quando aggiungi un gestore dell'identità digitale (IdP) Google.

Come registrare un'app con Apple

Per la maggior parte delle up-to-date informazioni sulla configurazione dell'accesso con Apple, consulta Configurazione dell'ambiente per l'accesso con Apple nella documentazione per gli sviluppatori Apple.

1. Creazione di un account sviluppatore con Apple.
2. Accedi con le tue credenziali Apple.
3. Sulla barra di navigazione a sinistra, scegli Certificates, Identifiers & Profiles (Certificati, identificatori e profili).
4. Nel riquadro di navigazione a sinistra, scegli Identifiers (Identificatori).
5. Nella pagina Identifiers (Identificatori), scegli l'icona +.
6. Nella pagina Register a New Identifier (Registra un nuovo identificatore), scegli App IDs (ID app), quindi scegli Continue (Continua).
7. Nella pagina Select a type (Seleziona un tipo), scegli App, quindi Continue (Continua).
8. Nella pagina Register an App ID (Registra un'ID app), procedi come indicato di seguito:

    1. In Description (Descrizione), inserisci una descrizione.

2. In App ID Prefix (Prefisso ID app), inserisci un Bundle ID ID bundle. Prendi nota del valore alla voce Prefisso ID app. Ne avrai bisogno per configurare Apple come provider di identità in Fase 2: aggiunta di un IdP social al bacino d'utenza.
3. In Funzionalità, scegli Accedi con Apple, quindi scegli Modifica.
4. Nella pagina Sign in with Apple: App ID Configuration (Accedi con Apple: configurazione ID app) scegli di configurare l'app come app primaria o raggruppata con altri ID app, quindi scegli Save (Salva).
5. Scegli Continua.

9. Nella pagina Confirm your App ID (Conferma l'ID app), scegli Register (Registra).
10. Nella pagina Identifiers (Identificatori), scegli l'icona +.
11. Nella pagina Register a New Identifier (Registra un nuovo identificatore), seleziona Services IDs (ID servizi), quindi scegli Continue (Continua).
12. Nella pagina Register a Services ID (Registra un ID servizi), procedi nel modo seguente:

    1. In Description (Descrizione), digitare una descrizione.
    2. In Identifier (Identificatore), digitare un identificatore. Prendi nota dell'ID servizi poiché ne avrai bisogno per configurare Apple come provider di identità in Fase 2: aggiunta di un IdP social al bacino d'utenza.
    3. Scegli Continue (Continua), quindi scegli Register (Registra).
13. Scegli l'ID servizi appena creato dalla pagina Identificatori.

    1. Scegli Sign In with Apple (Accedi con Apple), quindi scegli Configure (Configura).
    2. Nella pagina Web Authentication Configuration (Configurazione autenticazione Web), seleziona l'ID dell'app creato in precedenza come Primary App ID (ID app principale).
    3. Scegli l'icona + accanto a Website URLs (URL siti Web).
    4. In Domains and subdomains (Domini e sottodomini), inserisci il dominio del pool di utenti senza prefisso `https://`.

       ```
       mydomain.us-east-1.amazoncognito.com
       ```

    5. In Return URLs (URL restituiti), inserisci il percorso dell'endpoint `/oauth2/idpresponse` del dominio del pool di utenti.

       ```
       https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
       ```

6. Scegli Next (Successivo), quindi Done (Fatto). Non è necessario verificare il dominio.
7. Scegli Continua, quindi Salva.

14. Nel riquadro di navigazione a sinistra, scegli Keys (Chiavi).
15. Nella pagina Keys (Chiavi), scegli l'icona +.
16. Nella pagina Register a New Key, (Registra una nuova chiave) procedi nel modo seguente:

    1. Alla voce Key Name (nome chiave), digita un nome della chiave.
    2. Scegli Accedi con Apple, quindi scegli Configura.
    3. Nella pagina Configure Key (Configura chiave) seleziona l'ID dell'app creato in precedenza come Primary App ID (ID app principale). Selezionare Salva.
    4. Scegli Continua, quindi scegli Registra.
17. Nella pagina Download Your Key (download della chiave), scegli Download per scaricare la chiave privata, quindi prendi nota dell' ID chiave visualizzato, poi scegli Done (Fine). Avrai bisogno di questa chiave privata e del valore di Key ID (ID chiave) visualizzato in questa pagina dopo aver scelto Apple come provider di identità in Fase 2: aggiunta di un IdP social al bacino d'utenza.

Fase 2: aggiunta di un IdP social al bacino d'utenza

Per configurare un pool di utenti (social IdP) con AWS Management Console

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua la voce Federated sign-in (Accesso federato), quindi seleziona Add an identity provider (Aggiungi un provider di identità).
5. Scegli un provider di identità social: Facebook, Google, Login with Amazon o Accedi con Apple.
6. Scegli uno dei seguenti passaggi, in base al provider di identità social che hai selezionato:

    - Google e Login with Amazon: inserisci l'ID del client dell'app e il segreto client dell'app generato nella sezione precedente.
    - Facebook: inserisci l'ID del client dell'app e il segreto del client dell'app generato nella sezione precedente, quindi scegli una versione API (ad esempio, versione 2.12). Ti consigliamo di selezionare la versione più recente possibile in quanto ogni versione dell'API di Facebook

ha un ciclo di vita e una data di deprecazione. Gli ambiti e gli attributi di Facebook possono variare a seconda delle versioni dell'API. Ti consigliamo di testare l'accesso alla tua identità social con Facebook per assicurarti che la federazione funzioni come previsto.

- Accedi con Apple: inserisci l'ID servizio, l'ID team, l'ID chiave e la chiave privata generati nella sezione precedente.

7. Inserisci i nomi degli ambiti autorizzati che intendi utilizzare. Gli ambiti definiscono a quali attributi utente (ad esempio `name` e `email`) intendi accedere con l'app. Per Facebook, devono essere separati da virgole. Per Google e Login with Amazon, devono essere separati da spazi. Per Sign in with Apple, seleziona le caselle di controllo per gli ambiti cui desideri accedere.

Provider di identità social	Ambiti di esempio
Facebook	`public_profile, email`
Google	`profile email openid`
Login with Amazon	`profile postal_code`
Accedi con Apple	`email name`

All'utente dell'app viene richiesto il consenso a fornire questi attributi all'app. Per ulteriori informazioni sugli ambiti dei provider di social, consulta la documentazione di Google, Facebook, Login with Amazon e Accedi con Apple.

Nel caso di Accedi con Apple, di seguito sono riportati scenari utente in cui gli ambiti potrebbero non essere restituiti:

- Un utente finale rileva degli errori dopo aver chiuso la pagina di accesso di Apple (può derivare da errori interni di Amazon Cognito o da un errore di scrittura dello sviluppatore)
- L'identificatore dell'ID servizio viene utilizzato tra bacini d'utenza e/o altri servizi di autenticazione
- Uno sviluppatore aggiunge ambiti dopo che l'utente finale ha effettuato l'accesso in precedenza (non viene recuperata nessuna nuova informazione)
- Uno sviluppatore elimina l'utente e quindi l'utente effettua nuovamente l'accesso rimuovendo l'app dal profilo dell'ID Apple

8. Mappa gli attributi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, vedi Specificazione di mappature degli attributi del provider di identità per il pool di utenti.
9. Scegli Create (Crea).
10. Dalla scheda App client integration (integrazione client dell'app), scegli uno dei client dell'app nella lista e modifica le impostazioni dell'interfaccia utente ospitata. Aggiungi il nuovo provider di identità social al client di app in Identity providers (Provider di identità).
11. Scegli Save changes (Salva modifiche).

Fase 3: test della configurazione dell'IdP social

Puoi creare un URL di accesso utilizzando gli elementi delle due sezioni precedenti. Utilizzalo per testare la tua configurazione IdP social.

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

Per individuare il dominio, vai alla pagina della console Domain name (Nome dominio) del bacino d'utenza. Il valore di client_id è disponibile nella pagina App client settings Impostazioni client di applicazioni. Utilizza l'URL di callback per il parametro redirect_uri. Questo è l'URL della pagina a cui l'utente verrà reindirizzato dopo aver completato la procedura di autenticazione.

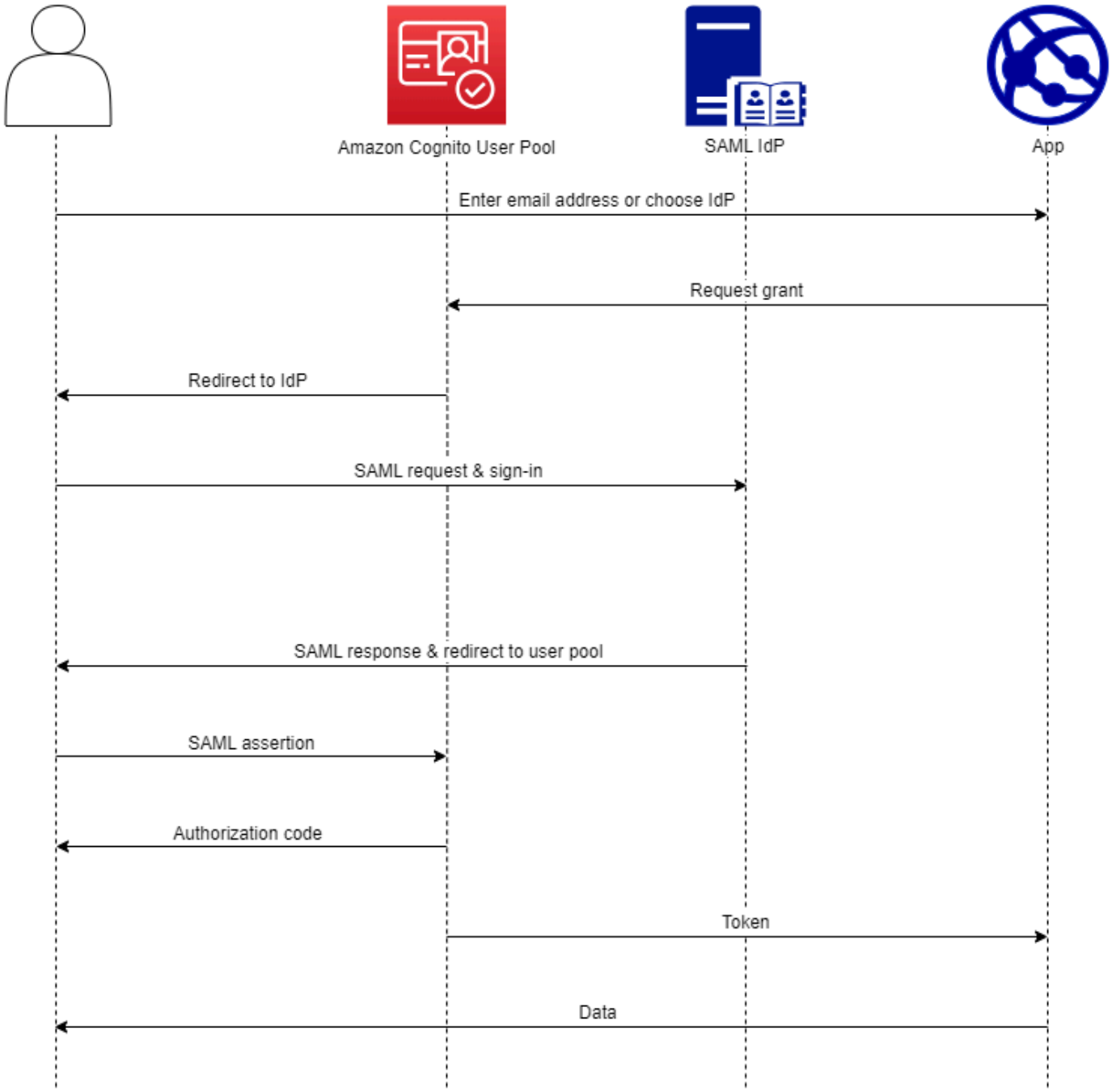
> **Note**
>
> Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro 5 minuti e reindirizza l'utente all'interfaccia utente ospitata. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

Utilizzo di provider di identità SAML con un pool di utenti

Puoi scegliere di fare in modo che gli utenti delle tue app Web e mobili accedano tramite un provider di identità SAML (IdP) come Microsoft Active Directory Federation Services (ADFS) o Shibboleth. È necessario selezionare un provider di identità SAML che supporti lo standard SAML 2.0.

Con l'interfaccia utente ospitata e gli endpoint federativi, Amazon Cognito autentica gli utenti IdP locali e di terze parti ed emette token web JSON (JWT). Con i token emessi da Amazon Cognito, puoi consolidare più fonti di identità in uno standard OpenID Connect (OIDC) universale per tutte le

tue app. Amazon Cognito può elaborare le asserzioni SAML dei tuoi provider di terze parti in quello standard SSO. Puoi creare e gestire un IdP SAML nell'API AWS Management Console dei pool di utenti di Amazon Cognito, tramite o con AWS CLI l'API dei pool di utenti. Per creare il tuo primo IdP SAML in, AWS Management Console consulta. Aggiungere e gestire provider di identità SAML in un pool di utenti



> **Note**
>
> La federazione con accesso tramite un IdP di terze parti è una funzionalità dei pool di utenti di Amazon Cognito. I pool di identità di Amazon Cognito, a volte chiamati identità federate di Amazon Cognito, sono un'implementazione della federazione che devi configurare separatamente in ogni pool di identità. Un pool di utenti può essere un IdP di terze parti per un pool di identità. Per ulteriori informazioni, consulta Pool di identità di Amazon Cognito.

Riferimento rapido per la configurazione IdP

Devi configurare il tuo IdP SAML per accettare richieste e inviare risposte al tuo pool di utenti. La documentazione per il tuo IdP SAML conterrà informazioni su come aggiungere il tuo pool di utenti come relying party o applicazione per il tuo IdP SAML 2.0. La documentazione che segue fornisce i valori da fornire per l'ID dell'entità SP e l'URL dell'assertion consumer service (ACS).

Riferimento rapido ai valori SAML del pool di utenti

ID dell'entità SP

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

URL ACS

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

È necessario configurare il pool di utenti per supportare il provider di identità. I passaggi di alto livello per aggiungere un IdP SAML esterno sono i seguenti.

1. Scarica i metadati SAML dal tuo IdP o recupera l'URL del tuo endpoint di metadati. Per informazioni, consulta Configurazione del provider di identità SAML di terze parti.
2. Aggiungi un nuovo IdP al tuo pool di utenti. Carica i metadati SAML o fornisci l'URL dei metadati. Per informazioni, consulta Aggiungere e gestire provider di identità SAML in un pool di utenti.
3. Assegna l'IdP ai client della tua app. Per informazioni, consultare Client dell'app pool di utenti.

Argomenti

- Cose da sapere su SAML IdPs nei pool di utenti di Amazon Cognito
- Distinzione tra maiuscole e minuscole dei nomi utente SAML
- Aggiungere e gestire provider di identità SAML in un pool di utenti
- Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito
- Utilizzo dell'accesso SAML avviato da SP
- Utilizzo dell'accesso SAML avviato da IdP
- Flusso di disconnessione SAML
- Firma e crittografia SAML
- Nomi e identificatori dei provider di identità SAML
- Configurazione del provider di identità SAML di terze parti

Cose da sapere su SAML IdPs nei pool di utenti di Amazon Cognito

Amazon Cognito elabora le asserzioni SAML per te

I pool di utenti di Amazon Cognito supportano la federazione SAML 2.0 con endpoint POST-binding. In questo modo, l'app non avrà bisogno di recuperare o analizzare le risposte dell'asserzione SAML poiché il bacino d'utenza riceve la risposta SAML direttamente dal provider di identità tramite un agente utente. Il bacino d'utenza si comporta da provider di servizi (SP) per conto della tua applicazione. Amazon Cognito supporta il single sign-on (SSO) avviato da SP e IdP come descritto nelle sezioni 5.1.2 e 5.1.4 della panoramica tecnica SAML V2.0.

Fornisci un certificato di firma IdP valido

Il certificato di firma nei metadati del tuo provider SAML non deve essere scaduto quando configuri l'IdP SAML nel tuo pool di utenti.

I pool di utenti supportano più certificati di firma

Quando l'IdP SAML include più di un certificato di firma nei metadati SAML, al momento dell'accesso il pool di utenti determina che l'asserzione SAML è valida se corrisponde a qualsiasi certificato nei metadati SAML. Ogni certificato di firma non deve contenere più di 4.096 caratteri.

Mantenere il parametro dello stato del relè

Amazon Cognito e l'IdP SAML gestiscono le informazioni sulla sessione con un parametro `relayState`.

1. Amazon Cognito supporta i valori `relayState` superiori a 80 byte. Mentre le specifiche SAML affermano che il valore `relayState` "non deve superare 80 byte di lunghezza", la pratica attuale del settore spesso si discosta da questo comportamento. Di conseguenza, il rifiuto `relayState` di valori superiori a 80 byte interromperà molte integrazioni di provider SAML standard.
2. Il `relayState` token è un riferimento opaco alle informazioni sullo stato gestite da Amazon Cognito. Amazon Cognito non garantisce i contenuti del parametro `relayState`. Non analizza i contenuti in modo tale che l'app dipende dal risultato. Per ulteriori informazioni consulta le Specifiche SAML 2.0.

Identifica l'endpoint ACS

Il provider di identità SAML richiede che venga impostato un endpoint assertion consumer. L'IdP reindirizza gli utenti a questo endpoint con la relativa asserzione SAML. Configura il seguente endpoint nel dominio del pool di utenti per il binding SAML 2.0 POST nel gestore dell'identità digitale 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
```

Per ulteriori informazioni sui domini del pool di utenti, consulta l'articolo Configurazione di un dominio di bacino d'utenza.

Nessuna asserzione ripetuta

Non puoi ripetere né riprodurre un'asserzione SAML sul tuo endpoint `saml2/idpresponse` Amazon Cognito. Un'asserzione SAML riprodotta dispone di un ID asserzione che duplica l'ID di una risposta IdP precedente.

L'ID del pool di utenti è l'ID dell'entità SP

È necessario fornire all'IdP l'ID del pool di utenti nel service provider (SP)`urn`, chiamato anche URI del pubblico o ID dell'entità SP. Il formato dell'URI audience per il pool di utenti è il seguente.

```
urn:amazon:cognito:sp:us-east-1_EXAMPLE
```

Puoi trovare l'ID del tuo pool di utenti nella sezione Panoramica del pool di utenti nella console Amazon Cognito.

Mappa tutti gli attributi obbligatori

Configura l'IdP SAML per fornire i valori per tutti gli attributi impostati come obbligatori nel pool di utenti. Ad esempio, `email` è un attributo obbligatorio comune per i pool di utenti. Prima che gli utenti possano effettuare l'accesso, le asserzioni IdP SAML devono includere un'attestazione mappata all'attributo pool di utenti `email`. Per ulteriori informazioni sulla mappatura attributi, consultareSpecificazione di mappature degli attributi del provider di identità per il bacino d'utenza.

Il formato di asserzione ha requisiti specifici

Il tuo IdP SAML deve includere le seguenti affermazioni nell'asserzione SAML.

1. Un reclamo. `NameID` Amazon Cognito associa un'asserzione SAML all'utente di destinazione tramite. `NameID` In caso di `NameID` modifiche, Amazon Cognito considera l'affermazione come riferita a un nuovo utente. L'attributo impostato `NameID` nella configurazione IdP deve avere un valore persistente.

```
<saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:persistent">
  carlos
</saml2:NameID>
```

2. Una richiesta `AudienceRestriction` con un valore `Audience` che imposta l'ID dell'entità SP del pool di utenti come destinazione della risposta.

```
<saml:AudienceRestriction>
  <saml:Audience> urn:amazon:cognito:sp:us-east-1_EXAMPLE
</saml:AudienceRestriction>
```

3. Per il single sign-on avviato da SP, un `Response` elemento con un `InResponseTo` valore dell'ID della richiesta SAML originale.

```
<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**
>
> Le asserzioni SAML avviate da IdP non devono contenere un valore. `InResponseTo`

4. Un `SubjectConfirmationData` elemento con un `Recipient` valore dell'`saml2/idpresponse`endpoint del pool di utenti e, per SAML avviato da SP, un valore che corrisponde all'ID della richiesta SAML originale. `InResponseTo`

```
<saml2:SubjectConfirmationData InResponseTo="_dd0a3436-bc64-4679-a0c2-
cb4454f04184" NotOnOrAfter="Date-time stamp" Recipient="https://mydomain.us-
east-1.amazoncognito.com/saml2/idpresponse"/>
```

Richieste di accesso avviate da SP

Quando Endpoint Authorize reindirizza l'utente alla pagina di accesso dell'IdP, Amazon Cognito include una richiesta SAML in un parametro URL della richiesta `HTTP GET`. Una richiesta SAML contiene informazioni sul tuo pool di utenti, incluso l'endpoint ACS. Facoltativamente, puoi applicare una firma crittografica a queste richieste.

Firma le richieste e crittografa le risposte

Ogni pool di utenti con un provider SAML genera una coppia di key pair asimmetrica e un certificato di firma per una firma digitale che Amazon Cognito assegna alle richieste SAML. Ogni IdP SAML esterno che configuri per supportare la risposta SAML crittografata fa sì che Amazon Cognito generi una nuova coppia di chiavi e un nuovo certificato di crittografia per quel provider. Per visualizzare e scaricare i certificati con la chiave pubblica, scegli il tuo IdP nella scheda Esperienza di accesso della console Amazon Cognito.

Per stabilire un rapporto di fiducia con le richieste SAML provenienti dal tuo pool di utenti, fornisci al tuo IdP una copia del certificato di firma SAML 2.0 del tuo pool di utenti. Il tuo IdP potrebbe ignorare le richieste SAML firmate dal tuo pool di utenti se non configuri l'IdP per accettare le richieste firmate.

1. Amazon Cognito applica una firma digitale alle richieste SAML che l'utente trasmette al tuo IdP. Il tuo pool di utenti firma tutte le richieste di single logout (SLO) e puoi configurare il tuo pool di utenti per firmare le richieste Single Sign-On (SSO) per qualsiasi IdP esterno SAML. Quando fornisci una copia del certificato, il tuo IdP può verificare l'integrità delle richieste SAML degli utenti.
2. Il tuo IdP SAML può crittografare le risposte SAML con il certificato di crittografia. Quando configuri un IdP con crittografia SAML, l'IdP deve inviare solo risposte crittografate.

Codifica caratteri non alfanumerici

Amazon Cognito non accetta caratteri UTF-8 a 4 byte come # o che il tuo IdP passa come valore di attributo. Puoi codificare il carattere in Base64, passarlo come testo e quindi decodificarlo nell'app.

Nell'esempio seguente, la richiesta di attributo non verrà accettata:

```
<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>
```

Contrariamente all'esempio precedente, la richiesta di attributo seguente verrà accettata:

```
<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>
```

L'endpoint di metadati deve avere una sicurezza valida a livello di trasporto

Se `InvalidParameterException` viene visualizzato durante la creazione di un provider di identità SAML con un URL di endpoint di metadati HTTPS, come "Error retrieving metadata from *<metadata endpoint>*" ("Errore recupero metadati da <endpoint di metedati>"), assicurati che l'endpoint di metadati abbia l'SSL configurato correttamente e che sia associato un certificato SSL valido. Per ulteriori informazioni sulla convalida dei certificati, consulta Che cos'è un certificato SSL/TLS? .

I client di app con SAML avviato da IdP possono accedere solo con SAML

Quando attivi il supporto per un IdP SAML 2.0 che supporta l'accesso avviato da IdP in un app client, puoi aggiungere solo altro IdPs SAML 2.0 a quell'app client. Ti viene impedito di aggiungere la directory utente nel pool di utenti e tutti i provider di identità esterni non SAML a un client di app configurato in questo modo.

Le risposte di disconnessione devono utilizzare l'associazione POST

L'`/saml2/logout`endpoint accetta richieste `LogoutResponse` come`HTTP POST`. I pool di utenti non accettano risposte di disconnessione con `HTTP GET` associazione.

Distinzione tra maiuscole e minuscole dei nomi utente SAML

Quando un utente federato tenta di accedere, il provider di identità SAML (IdP) trasmette un messaggio univoco ad Amazon `NameId` Cognito nell'asserzione SAML dell'utente. Amazon Cognito identifica un utente federato da SAML in base all'attestazione `NameId`. Indipendentemente dalle impostazioni di distinzione tra maiuscole e minuscole del tuo pool di utenti, Amazon Cognito riconosce un utente federato di ritorno da un IdP SAML quando trasmette la sua dichiarazione unica e con distinzione tra maiuscole e minuscole. `NameId` Se mappi un attributo come `email` a `NameId` e l'utente modifica il proprio indirizzo e-mail, non può accedere alla tua app.

Mappa `NameId` nelle asserzioni SAML da un attributo del provider di identità con valori che non cambiano.

Ad esempio, Carlos ha un profilo utente nel bacino d'utenza senza distinzione tra maiuscole e minuscole da un'asserzione SAML di Active Directory Federation Services (ADFS) che ha passato un valore `NameId` di `Carlos@example.com`. La prossima volta che Carlos tenta di accedere, il provider di identità ADFS passa un valore `NameId` di `carlos@example.com`. Poiché `NameId` deve rispettare esattamente la distinzione tra maiuscole e minuscole, l'accesso non ha esito positivo.

Se gli utenti non riescono ad accedere dopo la modifica di `NameID`, elimina i profili utente dal bacino d'utenza. Amazon Cognito crea nuovi profili utente la prossima volta che viene eseguito l'accesso.

Argomenti

- Aggiungere e gestire provider di identità SAML in un pool di utenti
- Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito
- Utilizzo dell'accesso SAML avviato da SP
- Utilizzo dell'accesso SAML avviato da IdP
- Flusso di disconnessione SAML
- Firma e crittografia SAML
- Nomi e identificatori dei provider di identità SAML
- Configurazione del provider di identità SAML di terze parti

Aggiungere e gestire provider di identità SAML in un pool di utenti

Le seguenti procedure mostrano come creare, modificare ed eliminare i provider SAML in un pool di utenti Amazon Cognito.

AWS Management Console

Puoi utilizzare il AWS Management Console per creare ed eliminare i provider di identità SAML (). IdPs

Prima di creare un IdP SAML, devi disporre del documento di metadati SAML che ottieni dall'IdP di terze parti. Per istruzioni su come ricevere o generare il documento di metadati SAML richiesto, consulta Configurazione del provider di identità SAML di terze parti.

Per configurare un provider di identità SAML 2.0 nel bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua la voce Federated sign-in (Accesso federato) quindi scegli Add an identity provider (Aggiungi un provider di identità).
5. Scegli un provider di identità SAML.
6. Inserisci il nome del provider. È possibile passare questo nome descrittivo in un parametro di `identity_provider` richiesta aEndpoint Authorize.
7. Inserisci gli Identifiers (Identificatori) separati da virgole. Un identificatore indica ad Amazon Cognito che dovrebbe controllare l'indirizzo e-mail inserito dall'utente al momento dell'accesso e quindi indirizzarlo al provider che corrisponde al proprio dominio.
8. Scegli Add sign-out flow (Aggiungi flusso di disconnessione) se desideri che Amazon Cognito invii richieste di disconnessione firmate al tuo provider quando un utente si disconnette. È necessario configurare il provider di identità SAML 2.0 per inviare le risposte di disconnessione all'endpoint `https://`*`mydomain.us-east-1.amazoncognito.com`*`/saml2/logout` creato al momento della configurazione dell'interfaccia utente ospitata. L'endpoint `saml2/logout` utilizza POST vincolanti.

> **Note**
>
> Se questa opzione è selezionata e il tuo IdP SAML prevede una richiesta di disconnessione firmata, devi anche fornire al tuo IdP SAML il certificato di firma dal tuo pool di utenti.

L'IdP SAML elabora la richiesta di disconnessione con firmata e disconnette l'utente dalla sessione Amazon Cognito.

9. Scegli la configurazione di accesso SAML avviata dall'IdP. Come best practice di sicurezza, scegli Accept only asserzioni SAML avviate da SP. Se hai preparato il tuo ambiente per accettare in modo sicuro sessioni di accesso SAML non richieste, scegli Accetta asserzioni SAML avviate da SP e IdP. Per ulteriori informazioni, consulta Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito.
10. Scegli una Metadata document source (Fonte del documento di metadati). Se il provider di identità offre metadati SAML a un URL pubblico, puoi scegliere l'opzione Metadata document URL (URL del documento di metadati) e inserire l'URL pubblico. In caso contrario, seleziona Upload metadata document (Carica documento di metadati) e seleziona un file di metadati scaricato dal tuo provider in precedenza.

> **Note**
>
> Ti consigliamo di inserire l'URL di un documento di metadati se il tuo provider ha un endpoint pubblico anziché caricare un file. Amazon Cognito aggiorna automaticamente i metadati dall'URL dei metadati. In genere, l'aggiornamento dei metadati avviene ogni 6 ore oppure prima della scadenza dei metadati, in base a ciò che avviene prima.

11. Mappa gli attributi tra il provider SAML e il pool di utenti per mappare gli attributi del provider SAML al profilo utente nel pool di utenti. Includi gli attributi richiesti del bacino d'utenza nella mappa degli attributi.

    Ad esempio, quando scegli User pool attribute (Attributo del bacino d'utenza) `email`, inserisci il nome dell'attributo SAML così come compare nell'asserzione SAML del provider di identità. Il provider di identità può offrire asserzioni SAML di esempio che possono essere di aiuto nell'individuare il nome. Alcuni IdPs usano nomi semplici, come`email`, mentre altri usano nomi come i seguenti.

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

12. Scegli Crea.

API/CLI

Utilizza i comandi seguenti per creare e gestire un provider di identità SAML.

Creazione di un provider di identità e caricamento di un documento di metadati

- AWS CLI: `aws cognito-idp create-identity-provider`

  Esempio con file di metadati: `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`

  Dove `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

  > **Note**
  >
  > Se <SAML metadata XML>contiene delle istanze del personaggio", devi aggiungere \ come carattere di escape:\".

  Esempio con URL di metadati: `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

Caricamento di un nuovo documento di metadati per un provider di identità

- AWS CLI: `aws cognito-idp update-identity-provider`

  Esempio con file di metadati: `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`

  Dove `details.json` contiene:

```
"ProviderDetails": {
      "MetadataFile": "<SAML metadata XML>",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

  > **Note**
  >
  > Se <SAML metadata XML>contiene delle istanze del personaggio", devi aggiungere \ come carattere di escape:\".

  Esempio con URL di metadati: `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

Recupero delle informazioni su un provider di identità specifico

- 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

Per elencare informazioni su tutti IdPs

- AWS CLI: `aws cognito-idp list-identity-providers`

  Esempio: `aws cognito-idp list-identity-providers --user-pool-id us-east-1_EXAMPLE --max-results 3`
- AWS API: ListIdentityProviders

Eliminazione di 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

Per configurare l'IdP SAML per l'aggiunta di un bacino d'utenza come relying party

- L'URN del provider di servizi del bacino d'utenza è: `urn:amazon:cognito:sp:us-east-1_EXAMPLE`. Amazon Cognito richiede un valore di restrizione del pubblico che corrisponda a questo URN nella risposta SAML. Configura il tuo IdP per utilizzare il seguente endpoint di binding POST per il messaggio di risposta da IdP a SP.

  ```
  https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse
  ```
- Il tuo IdP SAML deve `NameID` compilare tutti gli attributi obbligatori per il tuo pool di utenti nell'asserzione SAML. `NameID`viene utilizzato per identificare in modo univoco l'utente federato SAML nel pool di utenti. Il tuo IdP deve trasmettere l'ID del nome SAML di ogni utente in un formato coerente con distinzione tra maiuscole e minuscole. Qualsiasi variazione del valore dell'ID del nome utente crea un nuovo profilo utente.

Fornire un certificato di firma all'IDP SAML 2.0

- Per scaricare una copia della chiave pubblica da Amazon Cognito che il tuo IdP può utilizzare per convalidare le richieste di disconnessione SAML, scegli la scheda Esperienza di accesso del tuo pool di utenti, seleziona il tuo IdP e, in Visualizza certificato di firma, seleziona Scarica come .crt.

Puoi eliminare qualsiasi provider SAML configurato nel tuo bacino d'utenza con la console Amazon Cognito.

Come eliminare un provider SAML

1. Accedi alla console Amazon Cognito.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli la scheda Esperienza di accesso e individua l'accesso tramite Federated Identity Provider.
4. Seleziona il pulsante di opzione accanto al SAML IdPs che desideri eliminare.
5. Quando viene richiesto di confermare l'opzione Delete identity provider (Elimina provider di identità), inserisci il nome del provider SAML per confermare l'eliminazione, quindi scegli Delete (Elimina).

Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito

Amazon Cognito supporta il single sign-on (SSO) avviato dal service provider (avviato da SP) e l'SSO avviato da IdP. Come migliore pratica di sicurezza, implementa l'SSO avviato da SP nel tuo pool di utenti. La sezione 5.1.2 della Panoramica tecnica di SAML V2.0 descrive l'SSO avviato dal provider di servizi. Amazon Cognito è il provider di identità della tua app. L'app è il provider di servizi che recupera i token per gli utenti autenticati. Tuttavia, quando utilizzi un provider di identità di terze parti per autenticare gli utenti, Amazon Cognito diventa il provider di servizi. Quando gli utenti SAML 2.0 si autenticano con un flusso avviato da SP, devono sempre prima effettuare una richiesta ad Amazon Cognito e reindirizzare all'IdP per l'autenticazione.

Per alcuni casi d'uso aziendali, l'accesso alle applicazioni interne inizia da un segnalibro in un dashboard ospitato dal provider di identità aziendale. Quando un utente seleziona un segnalibro, il provider di identità genera una risposta SAML e la invia al provider di servizi per autenticare l'utente con l'applicazione.

Puoi configurare un IdP SAML nel tuo pool di utenti per supportare l'SSO avviato da IdP. Quando supporti l'autenticazione avviata da IdP, Amazon Cognito non può verificare di aver richiesto la risposta SAML che riceve perché Amazon Cognito non avvia l'autenticazione con una richiesta SAML. Nell'SSO avviato da SP, Amazon Cognito imposta parametri di stato che convalidano una risposta SAML rispetto alla richiesta originale. Con l'accesso avviato da SP puoi anche proteggerti dalla falsificazione delle richieste tra siti (CSRF).

Per un esempio di come creare SAML avviato da SP in un ambiente in cui non desideri che gli utenti interagiscano con l'interfaccia utente ospitata dal pool di utenti, consulta. Scenario di esempio: aggiungi ai preferiti le app Amazon Cognito in una dashboard aziendale

Argomenti

- Scenario di esempio: aggiungi ai preferiti le app Amazon Cognito in una dashboard aziendale

Scenario di esempio: aggiungi ai preferiti le app Amazon Cognito in una dashboard aziendale

Puoi creare segnalibri nelle dashboard SAML o OIDC IdP che forniscono ai pool di utenti di Amazon Cognito l'accesso SSO alle applicazioni Web. Puoi collegarti ad Amazon Cognito in modo da non richiedere agli utenti di accedere con l'interfaccia utente ospitata. A tale scopo, aggiungi un segnalibro di accesso al tuo portale che reindirizza al tuo pool di utenti Endpoint Authorize Amazon Cognito nel seguente formato.

```
https://mydomain.us-east-1.amazoncognito.com/authorize?
response_type=code&identity_provider=MySAMLIdP&client_id=1example23456789&redire
www.example.com
```

> **Note**
>
> È possibile utilizzare anche un parametro `idp_identifier` invece del parametro `identity_provider` nella richiesta all'endpoint di autorizzazione. Un identificatore IdP è un nome o dominio e-mail alternativo che puoi configurare quando crei un provider di identità nel tuo pool di utenti. Per informazioni, consulta Nomi e identificatori dei provider di identità SAML.

Quando usi i parametri appropriati nella richiesta a `/authorize`, Amazon Cognito inizia silenziosamente il flusso di accesso avviato dal provider di servizi e reindirizza l'utente all'accesso con il tuo provider di identità.

Per iniziare, aggiungi un IdP SAML nel tuo pool di utenti. Crea un client di app che utilizza il provider di identità SAML per l'accesso e ha l'URL dell'app come URL di callback autorizzato. Per ulteriori informazioni sui client di app, consulta Client dell'app pool di utenti.

Prima di distribuire questo accesso autenticato al portale, verifica l'accesso avviato da SP all'app dall'interfaccia utente ospitata. Per ulteriori informazioni su come configurare un provider di identità SAML in Amazon Cognito, consulta Configurazione del provider di identità SAML di terze parti.

Il diagramma seguente mostra un flusso di autenticazione che emula l'SSO avviato da provider di identità. Gli utenti possono autenticarsi con Amazon Cognito da un collegamento nel portale aziendale.

Dopo aver soddisfatto i requisiti, crea un segnalibro Endpoint Authorize che includa uno o un parametro. `identity_provider` `idp_identifier` L'autenticazione dell'utente procede come segue.

1. L'utente accede al pannello di controllo del provider di identità SSO. Le applicazioni per aziende a cui l'utente è autorizzato ad accedere popolano questo pannello di controllo.
2. L'utente sceglie il collegamento all'applicazione che esegue l'autenticazione con Amazon Cognito. In molti portali SSO è possibile aggiungere un collegamento all'app personalizzato. Qualsiasi caratteristica che puoi utilizzare per creare un collegamento a un URL pubblico funzionerà nel portale SSO.
3. Il collegamento all'app personalizzato nel portale SSO indirizza l'utente all'Endpoint Authorize del bacino d'utenza. Il collegamento include i parametri per `response_type`, `client_id`, `redirect_uri` e `identity_provider`. Il parametro `identity_provider` è il nome assegnato al provider di identità nel bacino d'utenza. Puoi anche utilizzare il parametro `idp_identifier` al posto del parametro `identity_provider`. Un utente accede all'endpoint di federazione da un link che contiene un `idp_identifier` parametro or. `identity_provider` Questo utente bypassa la pagina di accesso e si autentica direttamente con il provider di identità. Per ulteriori informazioni sulla denominazione IdPs SAML, consulta. Nomi e identificatori dei provider di identità SAML

   URL di esempio

   ```
   https://mydomain.us-east-1.amazoncognito.com/authorize?
   response_type=code&
   identity_provider=MySAMLIdP&
   client_id=1example23456789&
   ```

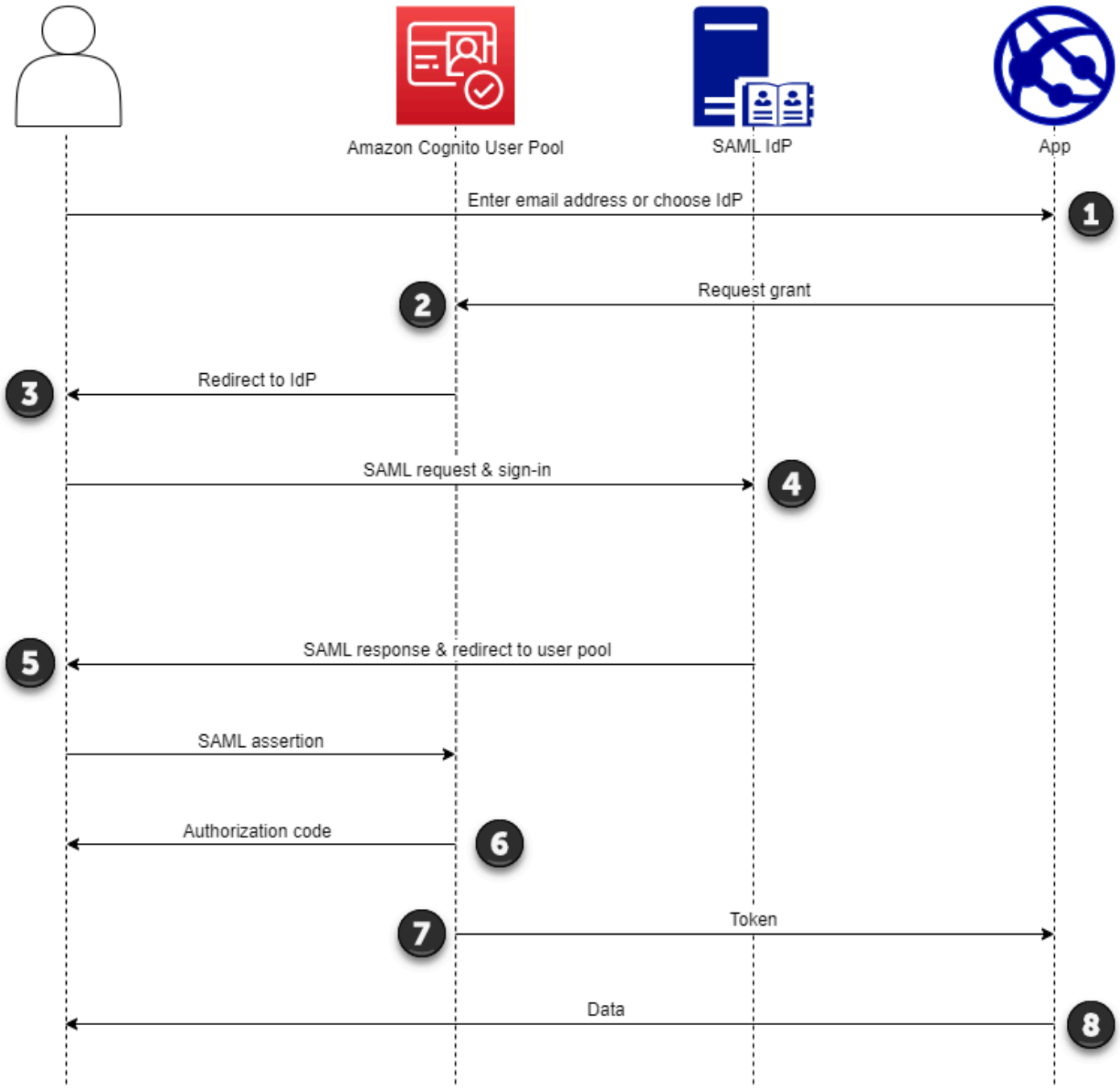
```
redirect_uri=https://www.example.com
```

4. Amazon Cognito reindirizza la sessione dell'utente al provider di identità con una richiesta SAML.
5. L'utente dovrebbe aver ricevuto un cookie di sessione dal provider di identità quando ha effettuato l'accesso al pannello di controllo. Il provider di identità utilizza questo cookie per convalidare l'utente in modo silenzioso e reindirizzarlo all'endpoint `idpresponse` di Amazon Cognito con una risposta SAML. Se non esiste una sessione attiva, il provider di identità autentica di nuovo l'utente prima di registrare la risposta SAML.
6. Amazon Cognito convalida la risposta SAML e crea o aggiorna il profilo utente in base all'asserzione SAML.
7. Amazon Cognito reindirizza l'utente all'app interna con un codice di autorizzazione. Hai configurato l'URL dell'app interna come URL di reindirizzamento autorizzato per il client di app.
8. L'app scambia il codice di autorizzazione con i token di Amazon Cognito. Per ulteriori informazioni, consulta Endpoint Token.

Utilizzo dell'accesso SAML avviato da SP

Come best practice, implementa l'accesso service-provider-initiated (avviato da SP) al tuo pool di utenti. Amazon Cognito avvia la sessione dell'utente e lo reindirizza al tuo IdP. Con questo metodo, hai il massimo controllo su chi presenta le richieste di accesso. Puoi anche consentire l'accesso avviato dall'IdP a determinate condizioni. Per ulteriori informazioni, consulta Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito.

La procedura seguente mostra come gli utenti accedono al tuo pool di utenti tramite un provider SAML.



1. L'utente inserisce il proprio indirizzo e-mail in una pagina di accesso. Per determinare il reindirizzamento dell'utente al suo IdP, puoi raccogliere il suo indirizzo email in un'app personalizzata o richiamare l'interfaccia utente ospitata nella visualizzazione web. Puoi configurare l'interfaccia utente ospitata per visualizzare un elenco IdPs o per richiedere solo un indirizzo e-mail.

2. L'app richiama l'endpoint di reindirizzamento del pool di utenti e richiede una sessione con l'ID client che corrisponde all'app e l'ID IdP che corrisponde all'utente.
3. Amazon Cognito reindirizza l'utente all'IdP con una richiesta SAML, facoltativamente firmata, in un elemento. `AuthnRequest`
4. L'IdP autentica l'utente in modo interattivo o con una sessione memorizzata in un cookie del browser.
5. L'IdP reindirizza l'utente all'endpoint di risposta SAML del pool di utenti con l'asserzione SAML crittografata facoltativamente nel payload POST.

   > **Note**
   >
   > Amazon Cognito annulla le sessioni che non ricevono una risposta entro 5 minuti e reindirizza l'utente all'interfaccia utente ospitata. Quando il tuo utente riscontra questo risultato, riceve un messaggio di errore. `Something went wrong`

6. Dopo aver verificato l'asserzione SAML e aver mappato gli attributi utente dalle affermazioni nella risposta, Amazon Cognito crea o aggiorna internamente il profilo dell'utente nel pool di utenti. In genere, il tuo pool di utenti restituisce un codice di autorizzazione alla sessione del browser dell'utente.
7. L'utente presenta il codice di autorizzazione all'app, che lo scambia con token web JSON (JWT).
8. L'app accetta ed elabora il token ID dell'utente come autenticazione, genera richieste autorizzate alle risorse con il relativo token di accesso e archivia il relativo token di aggiornamento.

Quando un utente si autentica e riceve una concessione di codice di autorizzazione, il pool di utenti restituisce ID, accesso e token di aggiornamento. Il token ID è un oggetto di autenticazione per la gestione delle identità basata su OIDC. Il token di accesso è un oggetto di autorizzazione con ambiti OAuth 2.0. Il token di aggiornamento è un oggetto che genera nuovi ID e token di accesso quando i token correnti dell'utente sono scaduti. Puoi configurare la durata dei token degli utenti nel client dell'app del pool di utenti.

Puoi anche scegliere la durata dei token di aggiornamento. Dopo la scadenza del token di aggiornamento, l'utente deve effettuare nuovamente l'accesso. Se si sono autenticati tramite un IdP SAML, la durata della sessione degli utenti è impostata dalla scadenza dei loro token, non dalla scadenza della sessione con il loro IdP. L'app deve archiviare il token di aggiornamento di ogni utente e rinnovare la sessione alla scadenza. L'interfaccia utente ospitata mantiene le sessioni utente in un cookie del browser valido per 1 ora.

Utilizzo dell'accesso SAML avviato da IdP

Quando configuri il tuo provider di identità per l'accesso a SAML 2.0 avviato da IdP, puoi presentare asserzioni SAML all'`saml2/idpresponse`endpoint nel dominio del tuo pool di utenti senza la necessità di avviare la sessione presso. Endpoint Authorize Un pool di utenti con questa configurazione accetta asserzioni SAML avviate da IdP da un provider di identità esterno del pool di utenti supportato dal client dell'app richiesto. I passaggi seguenti descrivono il processo generale di configurazione e accesso con un provider SAML 2.0 avviato da IdP.

1. Crea o designa un pool di utenti e un client per l'app.
2. Crea un IdP SAML 2.0 nel tuo pool di utenti.
3. Configura il tuo IdP per supportare l'avvio dell'IdP. SAML avviato da IdP introduce considerazioni sulla sicurezza a cui altri provider SSO non sono soggetti. Per questo motivo, non puoi aggiungere elementi non SAML IdPs, incluso il pool di utenti stesso, a nessun client di app che utilizza un provider SAML con accesso avviato da IdP.
4. Associa il tuo provider SAML avviato dall'IdP a un client di app nel tuo pool di utenti.
5. Indirizza l'utente alla pagina di accesso del tuo IdP SAML e recupera un'asserzione SAML.
6. Indirizza il tuo utente all'endpoint del tuo pool di utenti con la sua asserzione SAML`saml2/idpresponse`.
7. Ricevi token web JSON (JWT).

Per accettare asserzioni SAML non richieste nel tuo pool di utenti, devi considerarne l'effetto sulla sicurezza dell'app. È probabile che si verifichino tentativi di contraffazione delle richieste e CSRF quando si accettano richieste avviate da IdP. Sebbene il tuo pool di utenti non sia in grado di verificare una sessione di accesso avviata dall'IdP, Amazon Cognito convalida i parametri di richiesta e le asserzioni SAML.

Inoltre, l'asserzione SAML non deve contenere un `InResponseTo` reclamo e deve essere stata emessa nei 6 minuti precedenti.

Devi inviare richieste con SAML avviato da IdP al tuo. `/saml2/idpresponse` Per le richieste di autorizzazione dell'interfaccia utente avviate e ospitate da SP, devi fornire parametri che identifichino il client dell'app richiesta, gli ambiti, l'URI di reindirizzamento e altri dettagli come parametri della stringa di query nelle richieste. `HTTP GET` Per le asserzioni SAML avviate da IdP, tuttavia, i dettagli della richiesta devono essere formattati come `RelayState` parametri nel corpo di una richiesta.

`HTTP POST` Il corpo della richiesta deve contenere anche l'asserzione SAML come parametro. `SAMLResponse`

Di seguito è riportato un esempio di richiesta per un provider SAML avviato da 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

Per configurare un IdP per SAML avviato da IdP

1. Crea un pool di utenti, un client di app e un provider di identità SAML.
2. Dissocia tutti i provider di identità social e OIDC dal client dell'app, se ne sono associati.
3. Vai alla scheda Esperienza di accesso del tuo pool di utenti.
4. In Accedi con Federated Identity Provider, modifica o aggiungi un provider SAML.
5. In Accesso SAML avviato da IdP, scegli Accetta asserzioni SAML iniziate da SP e da IdP.
6. Seleziona Salvataggio delle modifiche.

API/CLI

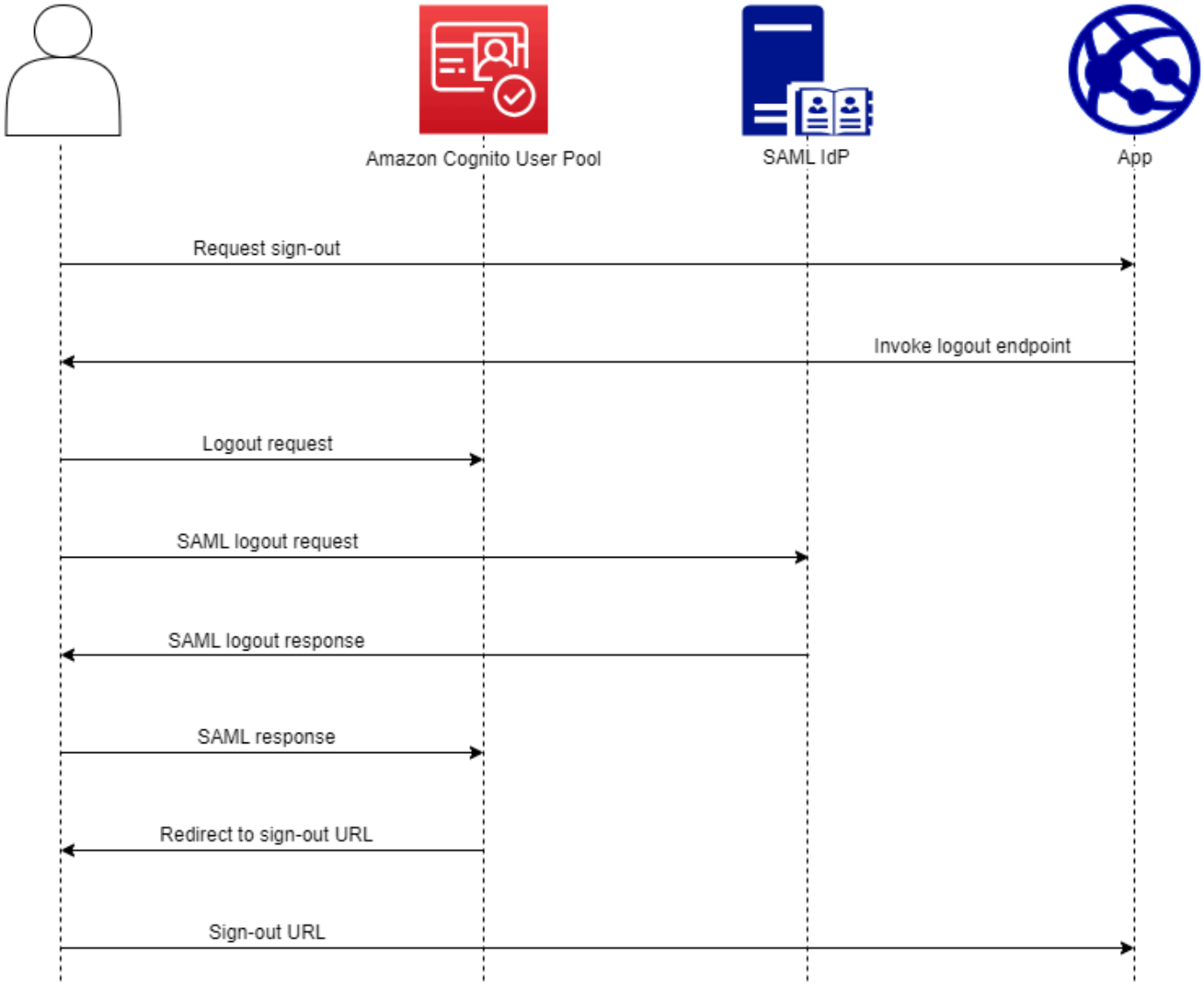
Per configurare un IdP per SAML avviato da IdP

Configura SAML avviato da IdP con il `IDPInit` parametro in una CreateIdentityProviderrichiesta o API. UpdateIdentityProvider Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta SAML avviato da IdP.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Flusso di disconnessione SAML

Amazon Cognito supporta il single logout SAML 2.0. Quando configuri il tuo IdP SAML per supportare il flusso di disconnessione, Amazon Cognito reindirizza l'utente con una richiesta di disconnessione SAML firmata al tuo IdP. Amazon Cognito determina la posizione di reindirizzamento dall'`SingleLogoutService`URL nei metadati del tuo IdP. Amazon Cognito firma la richiesta di disconnessione con il certificato di firma del tuo pool di utenti.



Quando indirizzi un utente con una sessione SAML all'`/logout`endpoint del tuo pool di utenti, Amazon Cognito reindirizza l'utente SAML con la seguente richiesta all'endpoint SLO specificato nei metadati 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]
```

L'utente torna quindi al tuo `saml2/logout` endpoint con un messaggio del suo `LogoutResponse` IdP. Il tuo IdP deve inviare una `HTTP POST` richiesta. `LogoutResponse` Amazon Cognito li reindirizza quindi alla destinazione di reindirizzamento indicata nella richiesta di disconnessione iniziale.

Il tuo provider SAML potrebbe inviare un messaggio contenente più `LogoutResponse` di uno. `AuthnStatement` Il `sessionIndex` valore inserito per primo `AuthnStatement` in una risposta di questo tipo deve corrispondere a quello contenuto `sessionIndex` nella risposta SAML che originariamente ha autenticato l'utente. Se si `sessionIndex` trova in un'altra`AuthnStatement`, Amazon Cognito non riconoscerà la sessione e l'utente non verrà disconnesso.

AWS Management Console

Per configurare la disconnessione SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.
2. Quando crei o modifichi il tuo provider di identità SAML, in Informazioni sul provider di identità, seleziona la casella con il titolo Aggiungi flusso di disconnessione.
3. Dalla scheda Esperienza di accesso del tuo pool di utenti, in Accesso tramite provider di identità federato, scegli il tuo IdP e individua il certificato di firma.
4. Scegli Scarica come .crt.
5. Configura il tuo provider SAML per supportare il single logout SAML e la firma delle richieste, quindi carica il certificato di firma del pool di utenti. Il tuo IdP deve reindirizzare verso il dominio del tuo `/saml2/logout` pool di utenti.

API/CLI

Per configurare la disconnessione SAML

Configura il logout singolo con il `IDPSignout` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta il single logout SAML.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",,
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
```

```
}
```

Firma e crittografia SAML

Amazon Cognito supporta richieste SAML firmate e risposte SAML crittografate per l'accesso e la disconnessione. Tutte le operazioni crittografiche durante le operazioni SAML del pool di utenti devono generare firme e testo cifrato con le chiavi generate da user-pool-provided Amazon Cognito. Attualmente, non è possibile configurare un pool di utenti per firmare richieste o accettare asserzioni crittografate con una chiave esterna.

> **Note**
>
> I certificati del pool di utenti sono validi per 10 anni. Una volta all'anno, Amazon Cognito genera nuovi certificati di firma e crittografia per il tuo pool di utenti. Amazon Cognito restituisce il certificato più recente quando richiedi il certificato di firma e firma le richieste con il certificato di firma più recente. Il tuo IdP può crittografare le asserzioni SAML con qualsiasi certificato di crittografia del pool di utenti non scaduto. I certificati precedenti continuano a essere validi per tutta la loro durata. Come best practice, aggiorna annualmente il certificato nella configurazione del provider.

Argomenti

- Accettazione di risposte SAML crittografate dal tuo IdP
- Firma delle richieste SAML

Accettazione di risposte SAML crittografate dal tuo IdP

Amazon Cognito e il tuo IdP possono stabilire la riservatezza nelle risposte SAML quando gli utenti accedono e si disconnettono. Amazon Cognito assegna una coppia di chiavi RSA pubblica-privata e un certificato a ogni provider SAML esterno che configuri nel tuo pool di utenti. Quando abiliti la crittografia delle risposte per il provider SAML del tuo pool di utenti, devi caricare il certificato su un IdP che supporti le risposte SAML crittografate. La connessione del pool di utenti al tuo IdP SAML non funziona prima che l'IdP inizi a crittografare tutte le asserzioni SAML con la chiave fornita.

Di seguito è riportata una panoramica del flusso di un accesso SAML crittografato.

1. L'utente avvia l'accesso e sceglie il proprio IdP SAML.

2. Il tuo pool di utenti Endpoint Authorize reindirizza l'utente al suo IdP SAML con una richiesta di accesso SAML. Il tuo pool di utenti può facoltativamente accompagnare questa richiesta con una firma che abilita la verifica dell'integrità da parte dell'IdP. Quando desideri firmare richieste SAML, devi configurare il tuo IdP per accettare le richieste che il tuo pool di utenti ha firmato con la chiave pubblica nel certificato di firma.
3. L'IdP SAML accede al tuo utente e genera una risposta SAML. L'IdP crittografa la risposta con la chiave pubblica e reindirizza l'utente all'endpoint del pool di utenti. `/saml2/idpresponse` L'IdP deve crittografare la risposta come definito dalla specifica SAML 2.0. Per ulteriori informazioni, vedere `Element <EncryptedAssertion>` in Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.
4. Il tuo pool di utenti decrittografa il testo cifrato nella risposta SAML con la chiave privata e accede all'utente.

> **Important**
>
> Quando abiliti la crittografia delle risposte per un IdP SAML nel tuo pool di utenti, l'IdP deve crittografare tutte le risposte con una chiave pubblica specifica per il provider. Amazon Cognito non accetta risposte SAML non crittografate da un IdP esterno SAML configurato per supportare la crittografia.

Qualsiasi IdP SAML esterno nel tuo pool di utenti può supportare la crittografia delle risposte e ogni IdP riceve la propria coppia di chiavi.

AWS Management Console

Per configurare la crittografia delle risposte SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.
2. Quando crei o modifichi il tuo provider di identità SAML, in Firma le richieste e crittografa le risposte, seleziona la casella con il titolo Richiedi asserzioni SAML crittografate da questo provider.
3. Dalla scheda Esperienza di accesso del tuo pool di utenti, in Accesso tramite provider di identità federato, seleziona il tuo IdP SAML e scegli Visualizza certificato di crittografia.
4. Scegli Scarica come .crt e fornisci il file scaricato al tuo IdP SAML. Configura il tuo IdP SAML per crittografare le risposte SAML con la chiave nel certificato.

API/CLI

Per configurare la crittografia delle risposte SAML

Configura la crittografia delle risposte con il `EncryptedResponses` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta la firma delle richieste.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Firma delle richieste SAML

La capacità di dimostrare l'integrità delle richieste SAML 2.0 al tuo IdP è un vantaggio in termini di sicurezza dell'accesso SAML avviato da Amazon Cognito SP. Ogni pool di utenti con un dominio riceve un certificato di firma X.509 del pool di utenti. Con la chiave pubblica in questo certificato, i pool di utenti applicano una firma crittografica alle richieste di disconnessione che il pool di utenti genera quando gli utenti selezionano un IdP SAML. Facoltativamente, puoi configurare il client dell'app per firmare le richieste di accesso SAML. Quando firmi le tue richieste SAML, il tuo IdP può verificare che la firma nei metadati XML delle tue richieste corrisponda alla chiave pubblica nel certificato del pool di utenti che fornisci.

AWS Management Console

Per configurare la firma delle richieste SAML

1. Crea un pool di utenti, un client di app e un IdP SAML.
2. Quando crei o modifichi il tuo provider di identità SAML, in Firma richieste e risposte crittografate, seleziona la casella con il titolo Firma le richieste SAML a questo provider.
3. Nella scheda Esperienza di accesso del tuo pool di utenti, in Accesso tramite provider di identità federato, scegli Visualizza certificato di firma.
4. Scegli Scarica come .crt e fornisci il file scaricato al tuo IdP SAML. Configura il tuo IdP SAML per verificare la firma delle richieste SAML in entrata.

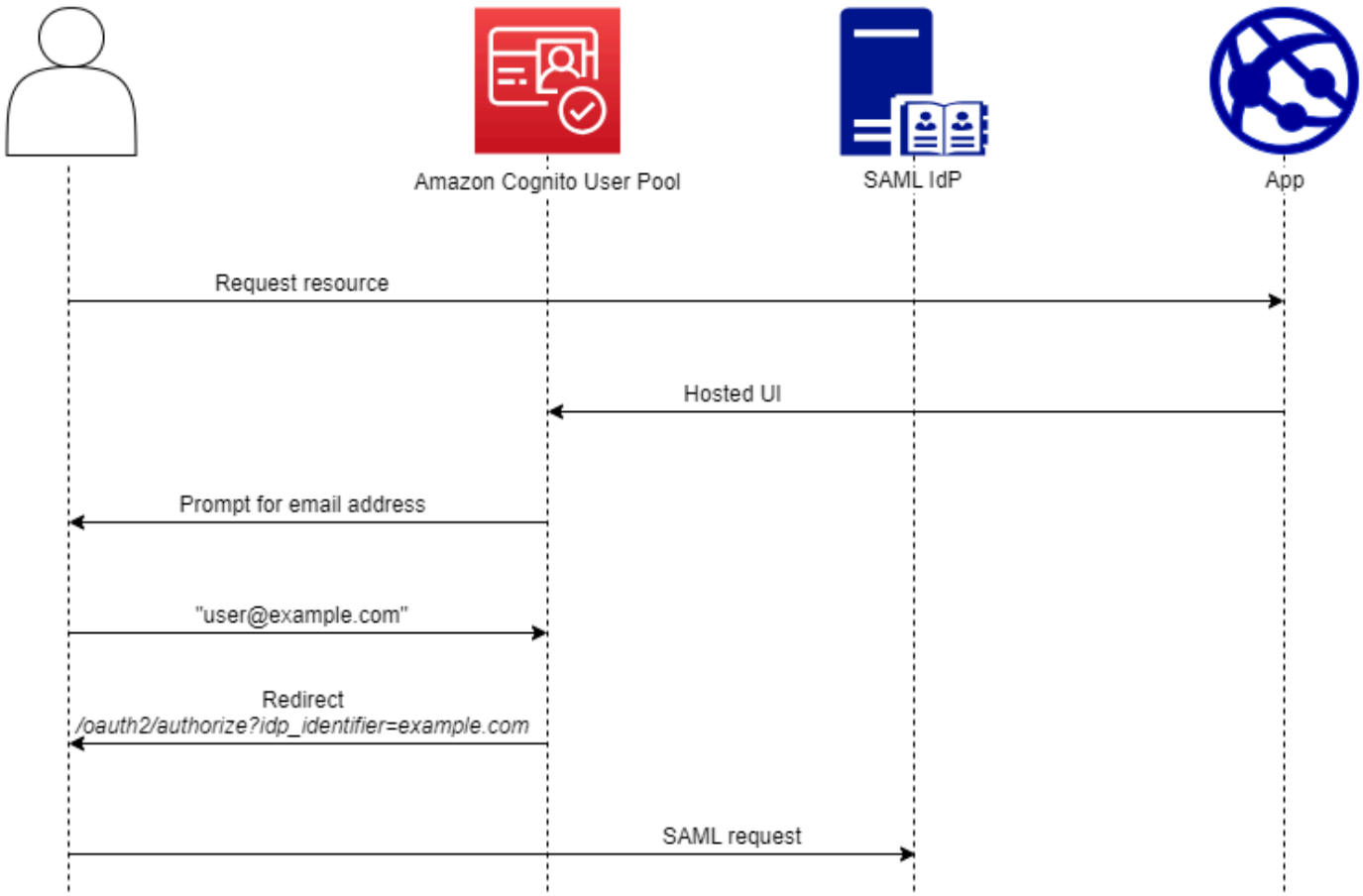
API/CLI

Per configurare la firma delle richieste SAML

Configura la firma delle richieste con il `RequestSigningAlgorithm` parametro di una richiesta CreateIdentityProvidero UpdateIdentityProviderAPI. Di seguito è riportato un esempio `ProviderDetails` di IdP che supporta la firma delle richieste.

```
"ProviderDetails": {
      "MetadataURL" : "https://myidp.example.com/saml/metadata",
      "IDPSignout" : "true",
      "RequestSigningAlgorithm" : "rsa-sha256",
      "EncryptedResponses" : "true",
      "IDPInit" : "true"
}
```

Nomi e identificatori dei provider di identità SAML



Quando dai un nome ai tuoi provider di identità SAML (IdPs) e assegni identificatori IdP, puoi automatizzare il flusso di richieste di accesso e disconnessione avviate da SP a quel provider. Per informazioni sui vincoli di stringa al nome del provider, consulta la proprietà di. `ProviderName` CreateIdentityProvider

Puoi anche scegliere fino a 50 identificatori per i tuoi provider SAML. Un identificatore è un nome descrittivo per un IdP nel pool di utenti e deve essere univoco all'interno del pool di utenti. Se gli identificatori SAML corrispondono ai domini e-mail degli utenti, l'interfaccia utente ospitata da Amazon Cognito richiede l'indirizzo e-mail di ciascun utente, valuta il dominio nel relativo indirizzo e-mail e lo reindirizza all'IdP corrispondente al dominio. Poiché la stessa organizzazione può possedere più domini, un singolo IdP può avere più identificatori.

Indipendentemente dal fatto che utilizzi o meno gli identificatori di dominio e-mail, puoi utilizzare gli identificatori in un'app multi-tenant per reindirizzare gli utenti all'IdP corretto. Se desideri ignorare completamente l'interfaccia utente ospitata, puoi personalizzare i link che presenti agli utenti in modo che reindirizzino Endpoint Authorize direttamente al loro IdP. Per accedere ai tuoi utenti con un identificatore e reindirizzarli al loro IdP, includi l'identificatore nel formato `idp_identifier=`*`myidp.example.com`* nei parametri di richiesta della loro richiesta di autorizzazione iniziale.

Un altro metodo per passare un utente al tuo IdP consiste nel compilare il parametro `identity_provider` con il nome del tuo IdP nel seguente formato URL.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&
identity_provider=MySAMLIdP&
client_id=1example23456789&
redirect_uri=https://www.example.com
```

Dopo che un utente ha effettuato l'accesso con il tuo IdP SAML, il tuo IdP lo reindirizza con una risposta SAML nel corpo dell'utente all'endpoint. `HTTP POST /saml2/idpresponse` Amazon Cognito elabora l'asserzione SAML e, se le affermazioni nella risposta soddisfano le aspettative, reindirizza all'URL di callback del client dell'app. Dopo aver completato l'autenticazione in questo modo, l'utente ha interagito con le pagine Web solo per il tuo IdP e la tua app.

Con gli identificatori IdP in un formato di dominio, l'interfaccia utente ospitata da Amazon Cognito richiede gli indirizzi e-mail al momento dell'accesso e quindi, quando il dominio e-mail corrisponde a un identificatore IdP, reindirizza gli utenti alla pagina di accesso del loro IdP. Ad esempio, crei un'app che richiede l'accesso da parte di dipendenti di due aziende diverse. La prima azienda,

AnyCompany A, possiede `exampleA.com` e`exampleA.co.uk`. La seconda società, AnyCompany B, possiede`exampleB.com`. Per questo esempio, ne hai impostate due IdPs, una per ogni società, come segue:

- Per il provider di identità A, puoi definire gli identificatori `exampleA.com` e `exampleA.co.uk`.
- Per il provider di identità B, puoi definire l'identificatore `exampleB.com`.

Nell'app, richiama l'interfaccia utente ospitata per il client dell'app per richiedere a ciascun utente di inserire il proprio indirizzo e-mail. Amazon Cognito ricava il dominio dall'indirizzo e-mail, correla il dominio a un IdP con un identificatore di dominio e reindirizza l'utente all'IdP corretto con una richiesta a quello che contiene un parametro di richiesta. Endpoint Authorize `idp_identifier` Ad esempio, se un utente entra`bob@exampleA.co.uk`, la pagina successiva con cui interagisce è la pagina di accesso IdP all'indirizzo. `https://auth.exampleA.co.uk/sso/saml`

Puoi anche implementare la stessa logica in modo indipendente. Nella tua app, puoi creare un modulo personalizzato che raccoglie l'input dell'utente e lo correla all'IdP corretto secondo la tua logica. Puoi generare portali di app personalizzati per ciascuno dei tenant dell'app, ognuno dei quali si collega all'endpoint di autorizzazione con l'identificatore del tenant nei parametri di richiesta.

Per raccogliere un indirizzo email e analizzare il dominio nell'interfaccia utente ospitata, assegna almeno un identificatore a ciascun IdP SAML che hai assegnato al client dell'app. Per impostazione predefinita, la schermata di accesso all'interfaccia utente ospitata mostra un pulsante per ciascuno dei pulsanti assegnati al IdPs client dell'app. Tuttavia, se hai assegnato correttamente gli identificatori, la pagina di accesso all'interfaccia utente ospitata è simile all'immagine seguente.

L'analisi del dominio nell'interfaccia utente ospitata richiede l'utilizzo di domini come identificatori IdP. Se assegni un identificatore di qualsiasi tipo a ciascun client SAML IdPs per un'app, l'interfaccia utente ospitata per quell'app non mostra più i pulsanti di selezione dell'IDP. Aggiungi identificatori IdP per SAML quando intendi utilizzare l'analisi delle e-mail o la logica personalizzata per generare reindirizzamenti. Se desideri generare reindirizzamenti silenziosi e desideri che l'interfaccia utente ospitata mostri un elenco di IdPs, non assegnare identificatori e utilizza il parametro di richiesta nelle tue richieste di autorizzazione. `identity_provider`

- Se assegni un solo provider di identità SAML al client di app, la pagina di accesso dell'interfaccia utente ospitata visualizza un pulsante per accedere con il provider di identità.
- Se assegni un identificatore a ogni IdP SAML che attivi per il client dell'app, nella pagina di accesso dell'interfaccia utente ospitata viene visualizzato un prompt di input per un indirizzo e-mail.

- Se ne hai più di uno IdPs e non assegni un identificatore a tutti, la pagina di accesso all'interfaccia utente ospitata mostra un pulsante per accedere con ogni IdP assegnato.
- Se hai assegnato degli identificatori al tuo IdPs e desideri che l'interfaccia utente ospitata mostri una selezione di pulsanti IdP, aggiungi un nuovo IdP privo di identificatore al client dell'app o crea un nuovo app client. Puoi anche eliminare un IdP esistente e aggiungerlo di nuovo senza un identificatore. Se crei un nuovo IdP, i tuoi utenti SAML creeranno nuovi profili utente. Questa duplicazione di utenti attivi potrebbe avere un impatto sulla fatturazione nel mese in cui modifichi la configurazione dell'IdP.

Per ulteriori informazioni sulla configurazione degli IdP, consulta Configurazione dei provider di identità per il bacino d'utenza.

Configurazione del provider di identità SAML di terze parti

Per configurare soluzioni di provider di identità (IdP) SAML 2.0 di terze parti in modo che funzionino con la federazione per i pool di utenti di Amazon Cognito, devi configurare il tuo IdP SAML per reindirizzare al seguente URL di Assertion Consumer Service (ACS):. `https://mydomain.us-east-1.amazoncognito.com/saml2/idpresponse` Se il pool di utenti ha un dominio Amazon Cognito, è possibile trovare il percorso del dominio del pool di utenti nella scheda App integration (Integrazione app) del pool di utenti nella console Amazon Cognito.

Alcuni SAML IdPs richiedono che tu fornisca il`urn`, chiamato anche URI del pubblico o ID di entità SP, nel modulo. `urn:amazon:cognito:sp:us-east-1_EXAMPLE` Puoi trovare l'ID del tuo pool di utenti nella sezione Panoramica del pool di utenti nella console Amazon Cognito.

Devi anche configurare il tuo IdP SAML per fornire valori per tutti gli attributi che hai designato come attributi obbligatori nel tuo pool di utenti. In genere, `email` è un attributo obbligatorio per i pool di utenti, nel qual caso l'IdP SAML deve fornire `email` una qualche forma di attestazione nella propria asserzione SAML e tu devi mappare l'attestazione all'attributo di quel provider.

Le seguenti informazioni di configurazione per le soluzioni IdP SAML 2.0 di terze parti sono un buon punto di partenza per configurare la federazione con i pool di utenti di Amazon Cognito. Per le informazioni più aggiornate, consulta direttamente la documentazione del tuo provider.

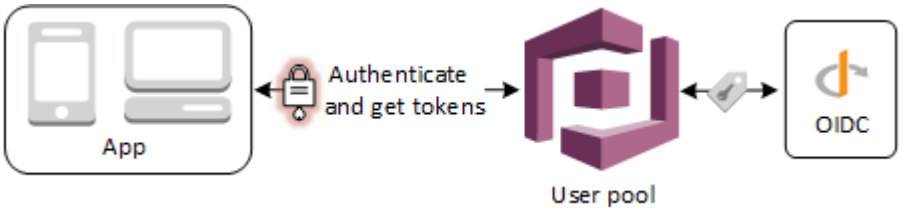
Per firmare le richieste SAML, devi configurare il tuo IdP in modo che consideri attendibili le richieste firmate dal certificato di firma del tuo pool di utenti. Per accettare risposte SAML crittografate, devi configurare il tuo IdP per crittografare tutte le risposte SAML nel tuo pool di utenti. Il tuo provider disporrà della documentazione sulla configurazione di queste funzionalità. Per un esempio di Microsoft, vedi Configurare la crittografia del token SAML di Microsoft Entra.

> **Note**
>
> Amazon Cognito richiede solo il documento di metadati del provider di identità. Il tuo provider potrebbe offrire informazioni di configurazione per la Account AWS federazione con SAML 2.0; queste informazioni non sono rilevanti per l'integrazione con Amazon Cognito.

Soluzione	Ulteriori informazioni
Microsoft Active Directory Federation Services (AD FS)	Federation Metadata Explorer
Okta	Come scaricare i metadati IdP e i certificati di firma SAML per l'integrazione di un'app SAML
Auth0	Configura Auth0 come provider di identità SAML
Ping Identity () PingFederate	Esportazione di metadati SAML da PingFederate
JumpCloud	Note di configurazione SAML
SecureAuth	Integrazione delle applicazioni SAML

Utilizzo di provider di identità OIDC con un pool di utenti

Puoi consentire ai tuoi utenti che dispongono già di account con provider di identità OpenID Connect (OIDC) (IdPs) di saltare la fase di registrazione e accedere all'applicazione utilizzando un account esistente. Con l'interfaccia utente Web ospitata integrata, Amazon Cognito fornisce la gestione dei token per gli utenti autenticati. In questo modo, i sistemi back-end possono standardizzare un set di token del bacino d'utenza.



> **Note**
>
> L'accesso tramite terze parti (federazione) è disponibile per i bacini d'utenza Amazon Cognito. Questa funzione è indipendente dalla federazione tramite pool di identità di Amazon Cognito (identità federate).

Puoi aggiungere un IdP OIDC al tuo pool di utenti nel AWS Management Console, tramite o con il metodo AWS CLI API del pool di utenti. CreateIdentityProvider

Argomenti

- Prerequisiti
- Fase 1: registrazione con un IdP OIDC
- Fase 2: Aggiunta di un IdP OIDC al bacino d'utenza
- Fase 3: Test della configurazione dell'IdP OIDC
- Flusso di autenticazione IdP del bacino d'utenza OIDC

Prerequisiti

Prima di iniziare, è necessario:

- Un bacino d'utenza con un client di applicazioni e dominio per il bacino d'utenza. Per ulteriori informazioni, consulta Creazione di un bacino d'utenza.
- Un IdP OIDC con la seguente configurazione:
  - Supporta l'autenticazione client `client_secret_post`. Amazon Cognito non controlla l'attestazione `token_endpoint_auth_methods_supported` nell'endpoint di rilevamento OIDC per il tuo IdP. Amazon Cognito non supporta l'autenticazione client `client_secret_basic`. Per ulteriori informazioni sull'autenticazione client, consulta Autenticazione client nella documentazione di OpenID Connect.
  - Utilizza HTTPS solo per endpoint OIDC come `openid_configuration`, `userInfo` e `jwks_uri`.
  - Utilizza solo le porte TCP 80 e 443 per gli endpoint OIDC.
  - Firma solo i token ID con algoritmi HMAC-SHA, ECDSA o RSA.
  - Pubblica un'attestazione `kid` dell'ID chiave presso il relativo `jwks_uri` e include un'attestazione `kid` nei rispettivi token.

Fase 1: registrazione con un IdP OIDC

Prima di creare un IdP OIDC con Amazon Cognito, è necessario registrare l'applicazione con l'IdP OIDC per ricevere un ID client e un segreto client.

Registrazione con un IdP OIDC

1. Crea un account sviluppatore con l'IdP OIDC.

   Collegamenti a OIDC IdPs

IdP OIDC	Installazione	URL di individuazione OIDC
Salesforce	Installa un provider di identità Salesforce	https://login.salesforce.com
Identità Ping	Installa un provider di identità Ping Identity	https://*il tuo indirizzo di dominio Ping*:9031/idp/userinfo.openid Ad esempio: `https://pf.company.com:9031/idp/userinfo.openid`
Okta	Installa un provider di identità Okta	https://*il tuo sottodominio Okta* .oktapreview.com oppure `https://`*`Your Okta subdomain`*`.okta.com`
Microsoft Azure Active Directory (AD)	Installa un provider di identità Microsoft Azure AD	`https://login.microsoftonline.com/`*`{tenant}`*`/v2.0`
Google	Installa un provider di identità Google	https://accounts.google.com **Note** Amazon Cognito offre Google come IdP di accesso social integrato

IdP OIDC	Installazione	URL di individuazione OIDC
		. È consigliabile utilizzare l'IdP integrato. Per informazioni, consulta Utilizzo di provider di identità social con un pool di utenti.

2. Registra l'URL del dominio del bacino d'utenza con l'`/oauth2/idpresponse`endpoint presso l'IdP OIDC. In questo modo verrà accettato dall'IdP OIDC durante l'autenticazione degli utenti da parte di Amazon Cognito.

```
https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
```

3. Registra l'URL di richiamata con il bacino d'utenza di Amazon Cognito. Questo è l'URL della pagina a cui l'utente viene reindirizzato da Amazon Cognito dopo aver completato la procedura di autenticazione.

```
https://www.example.com
```

4. Seleziona i tuoi scopes (ambiti). L'ambito openid (OpenID) è obbligatorio. L'ambito email (E-mail) è necessario per ottenere l'accesso alle richieste email ed email_verifiedhttps://openid.net/specs/openid-connect-basic-1_0.html#StandardClaims.
5. L'IdP OIDC fornisce un ID client e un segreto client. Puoi utilizzarli durante la configurazione di un IdP OIDC nel bacino d'utenza.

Esempio: utilizza Salesforce come IdP OIDC con il tuo bacino d'utenza

È possibile utilizzare un provider di identità OIDC per stabilire la fiducia tra un provider di identità compatibile con OIDC, ad esempio Salesforce, e il tuo bacino d'utenza.

1. Creazione di un account sul sito Web di Salesforce Developers.
2. Accedi tramite il tuo account sviluppatore impostato nella fase precedente.
3. Dalla pagina di Salesforce, procedi in una delle modalità seguenti:

   - Se utilizzi Lightning Experience, scegli l'icona a forma di ingranaggio delle impostazioni, quindi scegli Setup Home (Imposta home).

- Se utilizzi Salesforce Classic e vedi la voce Setup (Impostazioni) nell'intestazione dell'interfaccia utente, sceglila.
- Se utilizzi Salesforce Classic e non vedi la voce Setup (Impostazioni) nell'intestazione, scegli il tuo nome nella barra di navigazione in alto e scegli Setup (Impostazioni) dall'elenco a discesa.

4. Nella barra di navigazione a sinistra, scegli Company Settings (Impostazioni azienda).
5. Nella barra di navigazione, scegli Domain (Dominio), inserisci un dominio e seleziona Create (Crea).
6. Nella barra di navigazione a sinistra, vai a Platform Tools Strumenti piattaforma e scegli App.
7. Scegli App Manager Gestore app.
8. a. Scegli New connected app (Nuova app connessa).
   b. Completa i campi obbligatori.

      In Start URL (URL di avvio), inserisci un URL all'endpoint `/authorize` per il dominio del bacino d'utenza che accede con il provider di identità Salesforce. Quando gli utenti accedono all'app connessa, Salesforce li indirizza a questo URL per completare l'accesso. Quindi Salesforce reindirizza gli utenti all'URL di callback associato al client di app.

      ```
      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. Abilita OAuth settings (Impostazioni OAuth) e inserisci l'URL dell'endpoint `/oauth2/idpresponse` per il dominio del bacino d'utenza in Callback URL (URL di callback). Questo è l'URL da cui Salesforce emette il codice di autorizzazione che Amazon Cognito scambia con un token OAuth.

      ```
      https://mydomain.us-east-1.amazoncognito.com/oauth2/idpresponse
      ```

9. Seleziona i tuoi scopes (ambiti). È necessario includere openid per l'ambito. Per concedere l'accesso alle attestazioni email e email_verified aggiungi l'ambito email. Separa gli ambiti con gli spazi.
10. Scegli Create (Crea).

    In Salesforce, l'ID client viene chiamato Consumer Key (Chiave consumatore), mentre il segreto client è un Consumer Secret (Segreto consumatore). Prendi nota dell'ID client e del segreto client. Li utilizzerai nella sezione successiva.

Fase 2: Aggiunta di un IdP OIDC al bacino d'utenza

In questa sezione viene configurato il bacino d'utenza in modo che elabori le richieste di autenticazione basate su OIDC e provenienti da un IdP OIDC.

Aggiunta di un IdP OIDC (console Amazon Cognito)

Aggiunta di un IdP OIDC

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (Bacini d'utenza) dal menu di navigazione.
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua l'opzione Federated sign-in (Accesso federato) e seleziona Add an identity provider (Aggiungi un provider di identità).
5. Scegli un provider di identità OpenID Connect.
6. Inserisci un nome univoco nel campo Provider name (Nome provider).
7. Inserisci l'ID client che hai ricevuto dal tuo provider nel campo ID client (Client ID).
8. Inserisci il segreto client che hai ricevuto dal tuo provider nel campo Client secret (Segreto client).
9. Inserisci gli Authorized scopes (Ambiti autorizzati) per questo provider. Gli ambiti definiscono quali gruppi di attributi utente (ad esempio `name` e `email`) verranno richiesti dalla tua applicazione al tuo provider. Gli ambiti devono essere separati da spazi, secondo la specifica OAuth 2.0.

   All'utente viene richiesto il consenso a fornire questi attributi all'app.
10. Scegli un Attribute request method (Metodo della richiesta di attributo) per fornire ad Amazon Cognito il metodo HTTP (GET o POST) da utilizzare per recuperare i dettagli dell'utente dall'endpoint userInfo gestito dal tuo provider.
11. Scegli un Setup method (Metodo di impostazione) per recuperare gli endpoint OpenID Connect dall'opzione Auto fill through issuer URL (Riempimento automatico attraverso URL dell'emittente) o da Manual input (Inserimento manuale). Utilizza lo strumento Auto fill through issuer URL (Riempimento automatico dell'URL dell'emittente) quando il tuo provider ha un endpoint pubblico `.well-known/openid-configuration` in cui Amazon Cognito può recuperare gli URL degli endpoint `authorization`, `token`, `userInfo` e `jwks_uri`.
12. Inserisci gli URL dell'emittente o degli endpoint `authorization`, `token`, `userInfo`, e `jwks_uri` dal provider di IdP.

> **Note**
>
> L'URL deve iniziare con `https://` e non deve finire con una barra `/`. Solo i numeri di porta 443 e 80 possono essere utilizzati con questo URL. Ad esempio, Salesforce usa questo URL:
> `https://login.salesforce.com`
> Scegliendo il riempimento automatico, il documento di individuazione deve utilizzare HTTPS per i seguenti valori: `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint`, e `jwks_uri`. In caso contrario, l'accesso avrà esito negativo.

13. Per impostazione predefinita, la richiesta OIDC sub viene mappata all'attributo del bacino d'utenza Username. Puoi mappare altre richieste OIDC agli attributi del bacino d'utenza. Inserisci la richiesta OIDC e scegli l'attributo del bacino d'utenza corrispondente dall'elenco a discesa. Ad esempio, l'indirizzo e-mail della richiesta viene spesso mappato all'attributo del bacino d'utenza Email (E-mail).
14. Mappa gli attributi dal provider di identità al bacino d'utenza. Per ulteriori informazioni, vedi Specificazione di mappature degli attributi del provider di identità per il pool di utenti.
15. Scegli Create (Crea).
16. Dalla scheda App client integration (integrazione client dell'app), scegli uno dei client dell'app nella lista e modifica le impostazioni dell'interfaccia utente ospitata. Aggiungi il nuovo provider di identità OIDC al client di app in Identity providers (Provider di identità).
17. Scegli Save changes (Salva modifiche).

Aggiunta di un IdP OIDC (AWS CLI)

- Vedi le descrizioni dei parametri per il metodo 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
```

Utilizza questa mappa di dettagli dei provider:

```
{
  "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"
}
```

Fase 3: Test della configurazione dell'IdP OIDC

È possibile creare l'URL di autorizzazione utilizzando gli elementi delle due sezioni precedenti e usandoli per testare la configurazione IdP OIDC.

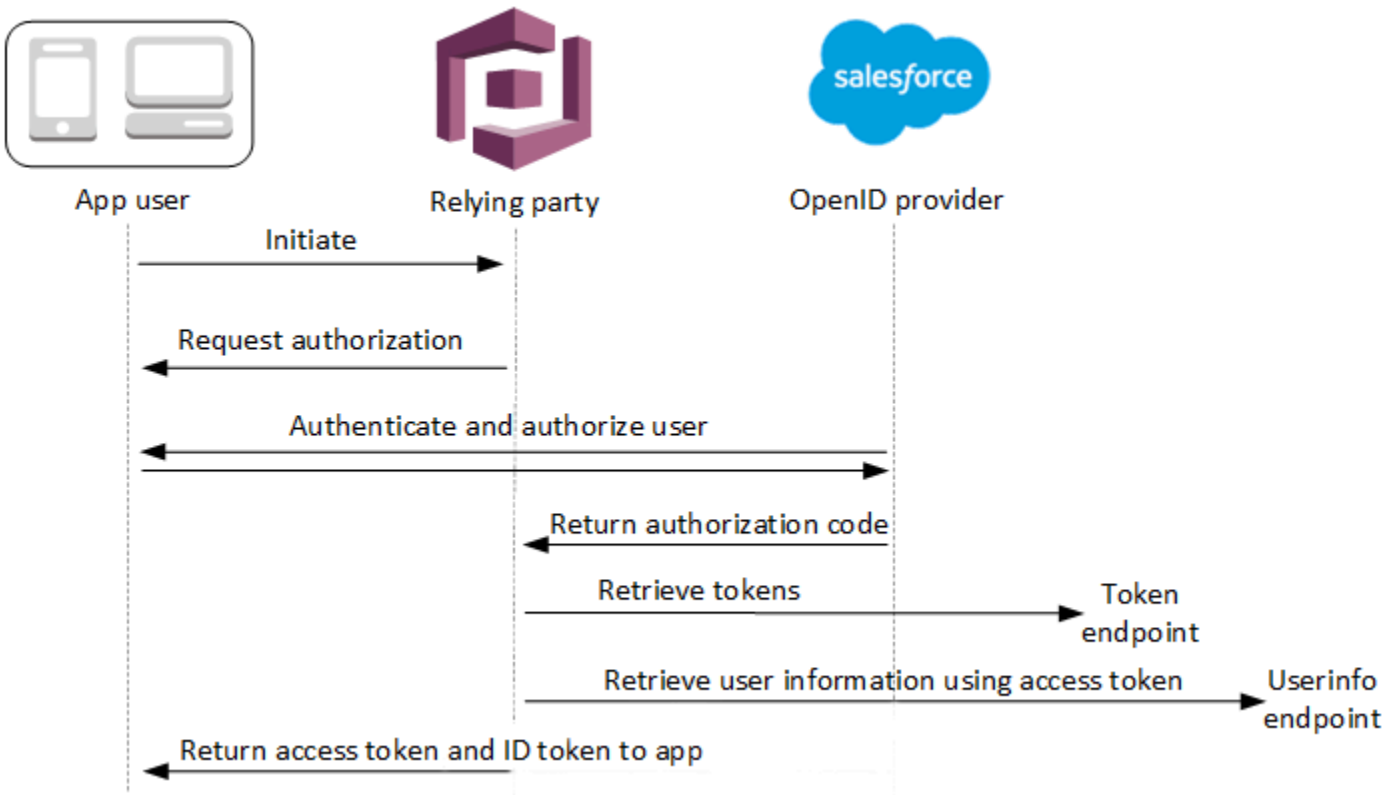
```
https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
response_type=code&client_id=1example23456789&redirect_uri=https://www.example.com
```

Per individuare il dominio, vai alla pagina della console Domain name (Nome dominio) del bacino d'utenza. Il client_id si trova nella pagina General settings (Impostazioni generali). Utilizza l'URL di callback per il parametro redirect_uri. Questo è l'URL della pagina a cui l'utente verrà reindirizzato dopo aver completato la procedura di autenticazione.

Flusso di autenticazione IdP del bacino d'utenza OIDC

Quando un utente accede alla tua applicazione utilizzando un IdP OIDC, passano attraverso il seguente flusso di autenticazione.

1. L'utente visualizza la pagina di accesso integrata di Amazon Cognito e visualizza l'opzione per accedere tramite un IdP OIDC, ad esempio Salesforce.
2. L'utente viene reindirizzato all'endpoint `authorization` del provider di identità OIDC.
3. Dopo che l'utente è stato autenticato, l'IdP OIDC viene reindirizzato a Amazon Cognito con un codice di autorizzazione.
4. Amazon Cognito scambia il codice di autorizzazione con l'IdP OIDC per un token di accesso.
5. Amazon Cognito crea o aggiorna l'account utente nel bacino d'utenza.
6. Amazon Cognito genera i token di connessione dell'applicazione, che potrebbero includere i token di identità, accesso e aggiornamento.



> **Note**
>
> Amazon Cognito annulla le richieste di autenticazione che non vengono completate entro 5 minuti e reindirizza l'utente all'interfaccia utente ospitata. Viene visualizzato il messaggio di errore `Something went wrong` nella pagina.

OIDC è un livello di identità che si aggiunge a OAuth 2.0, che specifica i token di identità in formato JSON (JWT) emessi dalle app client OIDC (relying party). IdPs Per informazioni su come aggiungere Amazon Cognito come relying party OIDC, consultare la documentazione dell'IdP OIDC.

Quando un utente esegue l'autenticazione con una concessione del codice di autorizzazione, il pool di utenti restituisce i token ID, di accesso e di aggiornamento. Il token ID è un token OIDC standard per la gestione delle identità, mentre il token di accesso è un token OAuth 2.0 standard. Per ulteriori informazioni sui tipi di concessione che il client app del pool di utenti può supportare, consulta Endpoint Authorize.

In che modo un pool di utenti elabora le richieste di un provider OIDC

Quando l'utente completa l'accesso con un provider OIDC di terze parti, l'interfaccia utente ospitata da Amazon Cognito recupera un codice di autorizzazione dal gestore dell'identità digitale. Il pool di utenti scambia il codice di autorizzazione per i token di accesso e ID con l'endpoint `token` del gestore dell'identità digitale. Il pool di utenti non passa questi token all'utente o all'app, ma li utilizza per creare un profilo utente con i dati resi disponibili nelle richieste nei propri token.

Amazon Cognito non convalida in modo indipendente il token di accesso. Richiede invece informazioni sugli attributi utente dall'endpoint `userInfo` del provider e si aspetta che la richiesta venga rifiutata se il token non è valido.

Amazon Cognito convalida il token ID del provider con i seguenti controlli:

1. Verifica che il provider abbia firmato il token con un algoritmo del seguente set: RSA, HMAC, crittografia a curva ellittica.
2. Se il provider ha firmato il token con un algoritmo di firma asimmetrico, verifica che l'ID della chiave di firma nella richiesta `kid` del token sia presente nell'endpoint `jwks_uri` del provider.
3. Confronta la firma del token ID con la firma prevista in base ai metadati del provider.
4. Confronta la richiesta `iss` con l'emittente OIDC configurata per il gestore dell'identità digitale.
5. Verifica che la richiesta `aud` corrisponda all'ID client configurato nel gestore dell'identità digitale o che contenga l'ID client configurato se sono presenti più valori nella richiesta `aud`.
6. Verifica che il timestamp nella richiesta `exp` non sia anteriore all'ora corrente.

Il pool di utenti convalida il token ID, quindi tenta una richiesta all'endpoint `userInfo` del provider con il token di accesso del provider. Recupera tutte le informazioni sul profilo utente che gli ambiti definiti nel token di accesso autorizzano a leggere. Il pool di utenti cerca quindi gli attributi utente impostati come obbligatori nel pool di utenti. Per gli attributi obbligatori è necessario creare

mappature degli attributi nella configurazione del provider. Il pool di utenti controlla il token ID del provider e la risposta `userInfo`. Il pool di utenti scrive tutte le richieste corrispondenti alle regole di mappatura negli attributi utente sul profilo utente del pool di utenti. Il pool di utenti ignora gli attributi che corrispondono a una regola di mappatura ma che non sono obbligatori e non sono presenti nelle richieste del provider.

Specificazione di mappature degli attributi del provider di identità per il bacino d'utenza

Puoi utilizzare l'API AWS Management Console, or the AWS CLI o, per specificare le mappature degli attributi per il provider di identità (IdP) del tuo pool di utenti.

Cose da sapere sulle mappature

Prima di iniziare a configurare la mappatura degli attributi utente, esamina i seguenti dettagli importanti.

- Quando un utente federato effettua l'accesso all'applicazione, per ogni attributo del pool di utenti richiesto deve essere presente una mappatura. Ad esempio, se il pool di utenti richiede un attributo `email` per l'accesso, mappa questo attributo al suo equivalente del gestore dell'identità digitale (IdP).
- Di default, gli indirizzi di posta elettronica mappati non sono verificati. Non è possibile verificare un indirizzo e-mail mappato utilizzando un codice una tantum. Tuttavia, puoi mappare un attributo del provider di identità per ottenere lo stato di verifica. Ad esempio, Google e la maggior parte dei provider OIDC includono l'attributo `email_verified`.
- Puoi mappare i token del gestore dell'identità digitale agli attributi personalizzati nel pool di utenti. I provider social visualizzano un token di accesso, mentre i provider OIDC visualizzano un token di accesso e ID. Per mappare un token, aggiungi un attributo personalizzato con una lunghezza massima di 2.048 caratteri, concedi al client dell'app l'accesso in scrittura all'attributo e mappa `access_token` o `id_token` dal gestore dell'identità digitale all'attributo personalizzato.
- Per ogni attributo del pool di utenti mappato, la lunghezza massima del valore (2.048 caratteri) deve essere sufficiente per il valore recuperato da Amazon Cognito dal gestore dell'identità digitale. In caso contrario, Amazon Cognito genera un errore quando gli utenti accedono alla tua applicazione. Amazon Cognito non supporta la mappatura tra token del gestore dell'identità digitale e attributi personalizzati quando i token hanno una lunghezza superiore a 2.048 caratteri.
- Amazon Cognito ricava l'`username`attributo nel profilo di un utente federato da affermazioni specifiche approvate dal tuo IdP federato, come illustrato nella tabella seguente. Amazon Cognito,

ad esempio, aggiunge questo valore di attributo al nome del tuo IdP. `MyOIDCIdP_[sub]` Se desideri che i tuoi utenti federati abbiano un attributo che corrisponda esattamente a un attributo nella tua directory utente esterna, mappalo a un attributo di accesso di Amazon Cognito come. `preferred_username`

Provider di identità	Attributo di origine **`username`**
Facebook	`id`
Google	`sub`
Login with Amazon	`user_id`
Accedi con Apple	`sub`
Provider SAML	`NameID`
Provider OpenID Connect (OIDC)	`sub`

- Amazon Cognito deve essere in grado di aggiornare gli attributi mappati del bacino d'utenza quando gli utenti accedono alla tua applicazione. Quando un utente accede tramite un provider di identità, Amazon Cognito aggiorna gli attributi mappati con le informazioni più recenti del provider di identità. Amazon Cognito aggiorna ciascun attributo mappato, anche se il valore corrente già soddisfa le informazioni più recenti. Per garantire che Amazon Cognito possa aggiornare gli attributi, controlla i seguenti requisiti:
  - Tutti gli attributi personalizzati del pool di utenti mappati dal gestore dell'identità digitale (IdP) devono essere mutabili. Puoi aggiornare gli attributi personalizzati modificabili in qualsiasi momento. Al contrario, puoi impostare un valore per un attributo personalizzato immutabile di un utente solo quando il profilo utente viene creato. Per creare un attributo personalizzato mutabile nella console Amazon Cognito, seleziona la casella di controllo Mutabile per l'attributo che aggiungi quando selezioni Aggiungi attributi personalizzati nella scheda Esperienza di registrazione. Oppure, se crei il tuo pool di utenti utilizzando l'operazione CreateUserPoolAPI, puoi impostare il `Mutable` parametro per ciascuno di questi attributi su. `true` Se il tuo IdP invia un valore per un attributo immutabile mappato, Amazon Cognito restituisce un errore e l'accesso non riesce.
  - Nelle impostazioni del client di applicazioni, gli attributi mappati devono essere scrivibili. Puoi impostare gli attributi scrivibili nella pagina Client di applicazioni nella console Amazon Cognito. In alternativa, se crei il client dell'app utilizzando l'operazione API `CreateUserPoolClient`,

puoi aggiungere questi attributi alla matrice `WriteAttributes`. Se il tuo IdP invia un valore per un attributo non scrivibile mappato, Amazon Cognito non imposta il valore dell'attributo e procede con l'autenticazione.

- Quando gli attributi IdP contengono più valori, Amazon Cognito semplifica tutti i valori in un'unica stringa delimitata da virgole e il modulo URL codifica i valori contenenti caratteri non alfanumerici (esclusi i caratteri '', '', 'e' ''). `.` `-` `*` `_` È necessario decodificare e analizzare i singoli valori prima di utilizzarli nell'app.

Specificazione di mappature degli attributi di provider di identità per il bacino d'utenza (AWS Management Console)

È possibile utilizzare AWS Management Console per specificare le mappature degli attributi per l'IdP del pool di utenti.

> **Note**
>
> Amazon Cognito eseguirà la mappatura delle registrazioni in ingresso agli attributi del bacino d'utenza solo se le registrazioni esistono nel token in ingresso. Se una registrazione mappata in precedenza non esiste più nel token in ingresso, non verrà eliminata o modificata. Se l'applicazione richiede la mappatura delle attestazioni eliminate, è possibile utilizzare il trigger Lambda di preautenticazione per eliminare l'attributo personalizzato durante l'autenticazione e consentire a questi attributi di ripopolarsi dal token in ingresso.

Specificazione della mappatura degli attributi di provider di identità social

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli la scheda Sign-in experience (Esperienza di accesso) e individua Federated sign-in (Accesso federato).
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure scegli il provider di identità Facebook, Google, Amazon o Apple che hai configurato. Individua Attribute mapping (Mappatura degli attributi) e scegli Edit (Modifica).

   Per ulteriori informazioni su come aggiungere un provider di identità social, consulta Utilizzo di provider di identità social con un pool di utenti.

5. Esegui i seguenti passaggi per ciascun attributo per cui intendi eseguire la mappatura:

    a. Seleziona un attributo dalla colonna User pool attribute (Attributo bacino d'utenza). Questo è l'attributo assegnato al profilo utente nel bacino d'utenza. Gli attributi personalizzati vengono elencati dopo gli attributi standard.

    b. Seleziona un attributo dalla colonna ***<provider>*** attribute (attributo del <provider>). Questo sarà l'attributo passato dalla directory del provider. Gli attributi noti del provider dei social sono forniti in un elenco a discesa.

    c. Per mappare attributi aggiuntivi tra il tuo IdP e Amazon Cognito, scegli Add another attribute (Aggiungi un altro attributo).

6. Scegli Save changes (Salva modifiche).

Specificazione di mappatura di attributo del provider SAML

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegli la scheda Sign-in experience (Esperienza di accesso) e individua Federated sign-in (Accesso federato).
4. Scegli l'opzione Add an identity provider (Aggiungi un provider di identità) oppure seleziona il provider di identità SAML configurato. Individua Attribute mapping (Mappatura degli attributi) e scegli Edit (Modifica). Per ulteriori informazioni su come aggiungere un provider di identità SAML, consulta Utilizzo di provider di identità SAML con un pool di utenti.
5. Esegui i seguenti passaggi per ciascun attributo per cui intendi eseguire la mappatura:

    a. Seleziona un attributo dalla colonna User pool attribute (Attributo bacino d'utenza). Questo è l'attributo assegnato al profilo utente nel bacino d'utenza. Gli attributi personalizzati vengono elencati dopo gli attributi standard.

    b. Seleziona un attributo dalla colonna SAML attribute (attributo SAML). Questo sarà l'attributo passato dalla directory del provider.

    Il provider di identità potrebbe offrire esempi di asserzioni SAML come riferimento. Alcuni IdPs utilizzano nomi semplici, ad esempio`email`, mentre altri utilizzano nomi di attributi in formato URL simili a:

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

c. Per mappare attributi aggiuntivi tra il tuo IdP e Amazon Cognito, scegli Add another attribute (Aggiungi un altro attributo).

6. Scegli Save changes (Salva modifiche).

Specificare le mappature degli attributi del provider di identità per il pool di utenti (e l'API)AWS CLIAWS

Utilizza i seguenti comandi per specificare le mappature degli attributi del provider di identità per il bacino d'utenza.

Specificazione di mappature degli attributi al momento della creazione del provider

- AWS CLI: `aws cognito-idp create-identity-provider`

  Esempio con file di metadati: `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`

  Dove `details.json` contiene:

```
{
    "MetadataFile": "<SAML metadata XML>"
}
```

> **Note**
>
> Se *<SAML metadata XML>* contiene virgolette ("), è necessario inserire il carattere di escape (\").

  Esempio con URL di metadati:

```
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

Specificazione delle mappature degli attributi per un provider di identità esistente

- AWS CLI: `aws cognito-idp update-identity-provider`

  Esempio: `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

Acquisizione di informazioni sulla mappatura degli attributi per un provider di identità specifico

- AWS CLI: `aws cognito-idp describe-identity-provider`

  Esempio: `aws cognito-idp describe-identity-provider --user-pool-id <user_pool_id> --provider-name <provider_name>`
- AWS API: DescribeIdentityProvider

Collegamento di utenti federati a un profilo utente esistente

Spesso, lo stesso utente ha un profilo con più provider di identità (IdPs) che hai collegato al tuo pool di utenti. Amazon Cognito può collegare ogni occorrenza di un utente allo stesso profilo utente nella directory. In questo modo, una persona che ha più utenti IdP può avere un'esperienza coerente nell'app. AdminLinkProviderForUserindica ad Amazon Cognito di riconoscere l'ID univoco di un utente nella tua directory federata come utente del pool di utenti. Un utente nel pool di utenti conta come un utente attivo mensile (MAU) ai fini della fatturazione quando hai zero o più identità federate associate al profilo utente.

Quando un utente federato accede al tuo pool di utenti per la prima volta, Amazon Cognito cerca un profilo locale che hai collegato alla sua identità. Se non esiste alcun profilo collegato, il tuo pool di utenti crea un nuovo profilo. Puoi creare un profilo locale e collegarlo al tuo utente federato in qualsiasi momento prima del primo accesso, in una richiesta `AdminLinkProviderForUser` API, in un'attività di preconfigurazione pianificata o in una. Trigger Lambda di pre-registrazione Dopo che l'utente ha effettuato l'accesso e Amazon Cognito ha rilevato un profilo locale collegato,

il pool di utenti legge le dichiarazioni dell'utente e le confronta con le regole di mappatura per l'IdP. Il tuo pool di utenti aggiorna quindi il profilo locale collegato con le attestazioni mappate in base al loro accesso. In questo modo, puoi configurare il profilo locale con le attestazioni di accesso e conservare le relative dichiarazioni di identità up-to-date presso il tuo provider. Dopo che Amazon Cognito ha associato il tuo utente federato a un profilo collegato, quest'ultimo accede sempre a quel profilo. Puoi quindi collegare più identità di provider del tuo utente allo stesso profilo, offrendo a un cliente un'esperienza coerente con la tua app. Per collegare un utente federato che ha precedentemente effettuato l'accesso, devi prima eliminare il suo profilo esistente. Puoi identificare i profili esistenti in base al loro formato: *`[Provider name]`*`_identifier`. Ad esempio, `LoginWithAmazon_amzn1.account.AFAEXAMPLE`. Un utente che hai creato e poi collegato a un'identità utente di terze parti ha lo stesso nome utente con cui è stato creato e un `identities` attributo che contiene i dettagli delle identità collegate.

> **Important**
>
> Poiché `AdminLinkProviderForUser` consente a un utente con un'identità federata esterna di accedere come utente esistente nel pool di utenti, è fondamentale che venga utilizzato solo con attributi esterni IdPs e del provider considerati attendibili dal proprietario dell'applicazione.

Ad esempio, se sei un fornitore di servizi gestiti (MSP) con un'app che condividi con più clienti. Ciascun cliente accede all'app tramite Active Directory Federation Services (ADFS). L'amministratore IT, Carlos, ha un account in ciascuno dei domini dei clienti. Vuoi che Carlos venga riconosciuto come amministratore dell'app ogni volta che accede, indipendentemente dal gestore dell'identità digitale (IdP).

Il tuo ADFS IdPs presenta l'indirizzo e-mail di Carlos `msp_carlos@example.com` nel `email` reclamo delle affermazioni SAML di Carlos ad Amazon Cognito. Crei quindi un utente nel pool di utenti con il nome utente.`Carlos`. I seguenti comandi AWS Command Line Interface (AWS CLI) collegano le identità di Carlos da ADFS1, ADFS2 e ADFS3. IdPs

> **Note**
>
> È possibile collegare un utente in base alle asserzioni di attributi specifici. Questa capacità è esclusiva di OIDC e SAML. IdPs Per altri tipi di gestori, è necessario collegarsi in base a un attributo di origine fisso. Per ulteriori informazioni, vedere. AdminLinkProviderForUser È necessario impostare `ProviderAttributeName` su `Cognito_Subject` quando colleghi

un IdP social a un profilo utente. `ProviderAttributeValue` deve essere l'identificatore univoco dell'utente con l'IdP.

```
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=msp_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=msp_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=msp_carlos@example.com
```

Il profilo utente `Carlos` nel pool di utenti ora dispone del seguente attributo `identities`.

```
[{
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS1",
    "providerType": "SAML",
    "issuer": "http://auth.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS2",
    "providerType": "SAML",
    "issuer": "http://auth2.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}, {
    "userId": "msp_carlos@example.com",
    "providerName": "ADFS3",
    "providerType": "SAML",
```

```
    "issuer": "http://auth3.example.com",
    "primary": false,
    "dateCreated": 111111111111111
}]
```

Informazioni importanti sul collegamento di utenti federati

- È possibile collegare fino a cinque utenti federati a ciascun profilo utente.
- È possibile collegare utenti federati a un profilo utente federato esistente o a un utente locale.
- Non è possibile collegare i provider ai profili utente in AWS Management Console.
- Il token ID dell'utente contiene tutti i gestori associati nell'asserzione `identities`.
- Puoi impostare una password per il profilo utente federato creato automaticamente in una richiesta API. AdminSetUserPassword Lo stato dell'utente cambia quindi da `EXTERNAL_PROVIDER` a `CONFIRMED`. Un utente in questo stato può accedere come utente federato e avviare flussi di autenticazione nell'API come un utente locale collegato. Possono anche modificare la password e gli attributi nelle richieste API autenticate da token come e. ChangePasswordUpdateUserAttributes Come best practice di sicurezza e per mantenere gli utenti sincronizzati con il tuo IdP esterno, non impostare la password per i profili di utenti federati. Collega invece gli utenti ai profili locali con `AdminLinkProviderForUser`.
- Amazon Cognito compila gli attributi utente in un profilo utente locale collegato quando l'utente accede tramite il proprio gestore dell'identità digitale. Amazon Cognito elabora le dichiarazioni di identità nel token ID di un IdP OIDC e controlla anche l'endpoint `userInfo` dei provider OAuth 2.0 e OIDC. Amazon Cognito assegna la priorità alle informazioni in un token ID rispetto alle informazioni di `userInfo`.

Quando scopri che il tuo utente non utilizza più un account utente esterno che hai collegato al suo profilo, puoi dissociare quell'account utente dal tuo utente del pool di utenti. Quando hai collegato il tuo utente, hai fornito il nome dell'attributo utente, il valore dell'attributo e il nome del provider nella richiesta. Per rimuovere un profilo di cui l'utente non ha più bisogno, effettua una richiesta AdminDisableProviderForUserAPI con parametri equivalenti.

Vedi AdminLinkProviderForUserla sintassi dei comandi e gli esempi aggiuntivi negli AWS SDK.

Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda

Amazon Cognito utilizza funzioni AWS Lambda per modificare il comportamento di autenticazione del tuo pool di utenti. Puoi configurare il tuo pool di utenti per richiamare automaticamente le funzioni Lambda prima della prima registrazione, dopo aver completato l'autenticazione e in diverse fasi intermedie. Le tue funzioni possono modificare il comportamento predefinito del flusso di autenticazione, effettuare richieste API per modificare il pool di utenti o altre risorse AWS e comunicare con sistemi esterni. Il codice delle funzioni Lambda è personale. Amazon Cognito invia i dati degli eventi alla tua funzione, attende che la funzione elabori i dati e nella maggior parte dei casi anticipa un evento di risposta che riflette le modifiche che desideri apportare alla sessione.

All'interno del sistema di eventi di richiesta e risposta, puoi introdurre problemi di autenticazione personalizzati, migrare gli utenti tra il tuo pool di utenti e un altro archivio di identità, personalizzare i messaggi e modificare i token web JSON (JWT).

I trigger Lambda possono personalizzare la risposta che Amazon Cognito restituisce all'utente dopo aver avviato un'operazione nel pool di utenti. Ad esempio, puoi impedire l'accesso da parte di un utente che altrimenti verrebbe consentito. Possono anche eseguire operazionidi runtime nell'ambiente AWS, nelle API esterne, nei database o negli archivi di identità. Il trigger di migrazione utente, ad esempio, può combinare un'operazione esterna con una modifica in Amazon Cognito, cercando le informazioni sull'utente in una directory esterna e quindi impostando gli attributi del nuovo utente in base alle informazioni esterne.

Quando un trigger Lambda è assegnato al pool di utenti, Amazon Cognito interrompe il flusso predefinito per richiedere informazioni alla funzione. Amazon Cognito genera pertanto un evento JSON e lo passa alla funzione. L'evento contiene informazioni sulla richiesta dell'utente di creare un account utente, eseguire l'accesso, reimpostare una password o aggiornare un attributo. La funzione ha quindi l'opportunità di eseguire l'operazione o di restituire l'evento senza modifiche.

La tabella riportata di seguito riepiloga alcuni dei modi in cui è possibile utilizzare i trigger Lambda per personalizzare le operazioni del bacino d'utenza:

Flusso di bacini d'utenza	Operazione	Descrizione
Flusso di autenticazione personalizzato	Definizione di una richiesta di autenticazione	Determina la prossima richiesta in un flusso di autorizzazione personalizzato

Flusso di bacini d'utenza	Operazione	Descrizione
	Creazione di una richiesta di autenticazione	Crea una richiesta in un flusso di autorizzazione personalizzato
	Verifica della risposta a una richiesta di autenticazione	Determina se una risposta è corretta in un flusso di autorizzazione personalizzato
Eventi di autenticazione	the section called “Trigger Lambda di pre-autenticazione”	Convalida personalizzata per accettare o rifiutare la richiesta di accesso
	the section called “Trigger Lambda di post-autenticazione”	Esegue il log di eventi per analisi personalizzate
	the section called “Trigger Lambda di pre-generazione del token”	Aumenta o sopprime le richieste di token
Registrazione	the section called “Trigger Lambda di pre-registrazione”	Esegue la convalida personalizzata accettando o negando la richiesta di registrazione
	the section called “Trigger Lambda di post-conferma”	Aggiunge messaggi di benvenuto personalizzati o il log di eventi per analisi personalizzate
	the section called “Trigger Lambda di migrazione utenti”	Migra un utente da una directory di utenti esistente ai bacini d'utenza
Messaggi	the section called “Trigger Lambda di messaggi personalizzati”	Esegue la personalizzazione avanzata e localizzazione di messaggi

Flusso di bacini d'utenza	Operazione	Descrizione
Creazione di token	the section called “Trigger Lambda di pre-generazione del token”	Aggiunge o rimuove attributi in ID token
Provider di terze parti di messaggi e-mail ed SMS	the section called “Trigger Lambda del mittente personalizzato”	Utilizza un provider di terze parti per inviare SMS e messaggi e-mail

Argomenti

- Considerazioni importanti
- Aggiunta di un trigger Lambda al bacino d'utenza
- Evento trigger Lambda per il bacino d'utenza
- Parametri comuni del trigger Lambda del bacino d'utenza
- Connessione delle operazioni API ai trigger Lambda
- Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti
- Trigger Lambda di pre-registrazione
- Trigger Lambda di post-conferma
- Trigger Lambda di pre-autenticazione
- Trigger Lambda di post-autenticazione
- Trigger Lambda di richieste di autenticazione personalizzate
- Trigger Lambda di pre-generazione del token
- Trigger Lambda di migrazione utenti
- Trigger Lambda di messaggi personalizzati
- Trigger Lambda del mittente personalizzato

Considerazioni importanti

Quando prepari i tuoi pool di utenti per le funzioni Lambda, considera quanto segue:

- Gli eventi inviati da Amazon Cognito ai trigger Lambda potrebbero cambiare con nuove funzionalità. Le posizioni degli elementi di risposta e richiesta nella gerarchia JSON potrebbero

cambiare o potrebbero essere aggiunti nomi di elementi. Nella funzione Lambda, puoi ricevere le coppie chiave-valore dell'elemento di input descritte in questa guida, ma una convalida dell'input più rigorosa può causare il mancato completamento delle funzioni.

- Puoi scegliere una di più versioni degli eventi inviati da Amazon Cognito ad alcuni trigger. Alcune versioni potrebbero richiedere di accettare una modifica ai prezzi di Amazon Cognito. Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon Cognito. Per personalizzare i token di accesso in un Trigger Lambda di pre-generazione del token, è necessario configurare il pool di utenti con funzionalità di sicurezza avanzate e aggiornare la configurazione del trigger Lambda per utilizzare la versione 2 dell'evento.
- Ad eccezione di Trigger Lambda del mittente personalizzato, Amazon Cognito richiama le funzioni Lambda in modo sincrono. Quando Amazon Cognito chiama la tua funzione Lambda, questa deve rispondere entro 5 secondi. In caso contrario e se la chiamata può essere ritentata, Amazon Cognito riprova la chiamata. Dopo 3 tentativi non riusciti, la funzione scade. Non puoi modificare questo valore di timeout di cinque secondi. Per ulteriori informazioni, consulta Modello di programmazione Lambda nella Guida per sviluppatori AWS Lambda.

  Amazon Cognito non riprova le chiamate di funzione che restituiscono un errore Invoke con un codice di stato HTTP 500-599. Questi codici indicano un problema di configurazione che impedisce a Lambda di avviare la funzione. Per ulteriori informazioni, consulta Gestione di errori e tentativi automatici in AWS Lambda.
- Non puoi dichiarare una versione della funzione nella configurazione del trigger Lambda. I pool di utenti di Amazon Cognito richiamano la versione più recente della tua funzione per impostazione predefinita. Tuttavia, puoi associare una versione della funzione a un alias e impostare il trigger `LambdaArn` sull'ARN dell'alias in una richiesta API CreateUserPool o UpdateUserPool. Questa opzione non è disponibile in AWS Management Console. Per ulteriori informazioni sugli alias, consulta Alias delle funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda.
- Se si elimina un trigger , è necessario aggiornare il corrispondente trigger nel bacino d'utenza. Ad esempio, se si elimina il trigger di post autenticazione, è necessario impostare il trigger Post authentication (Post autenticazione) nel bacino d'utenza corrispondente su none (nessuno).
- Se la funzione Lambda non restituisce i parametri di richiesta e risposta ad Amazon Cognito o restituisce un errore, l'evento di autenticazione non va a buon fine. Puoi restituire un errore nella funzione per impedire a un utente di eseguire la registrazione, l'autenticazione, la generazione di token o qualsiasi altra fase del flusso di autenticazione di un utente che richiami il trigger Lambda.

  L'interfaccia utente ospitata di Amazon Cognito restituisce gli errori generati dai trigger Lambda come testo di errore sopra la richiesta di accesso. L'API dei pool di utenti di Amazon Cognito

restituisce errori dei trigger nel formato *[trigger]* `failed with error` *[error text from response]*. Come best practice, genera nelle funzioni Lambda solo gli errori che desideri mostrare agli utenti. Usa metodi di output come `print()` per registrare eventuali informazioni sensibili o di debug in CloudWatch Logs. Per vedere un esempio, consulta Esempio di pre-registrazione: registrazione rifiutata se il nome utente ha meno di cinque caratteri.

- Puoi aggiungere una funzione Lambda in un altro Account AWS come un trigger per il pool di utenti. È necessario aggiungere trigger tra account con le operazioni API CreateUserPool e UpdateUserPool o i loro equivalenti in AWS CloudFormation e AWS CLI. Non puoi aggiungere funzioni tra account in AWS Management Console.
- Quando aggiungi un trigger Lambda nella console di Amazon Cognito, Amazon Cognito aggiunge alla funzione una policy basata sulle risorse che consente al pool di utenti di invocare la funzione. Quando crei un trigger Lambda al di fuori della console di Amazon Cognito, inclusa una funzione tra account, devi aggiungere le autorizzazioni alla policy basata sulle risorse della funzione Lambda. Le autorizzazioni aggiunte devono consentire ad Amazon Cognito di invocare la funzione per conto del pool di utenti. È possibile aggiungere autorizzazioni da Lambda Console oppure usare l'operazione API AddPermission di Lambda.

  Esempio di policy basate su risorse Lambda

  La seguente policy basata su risorse Lambda garantisce ad Amazon Cognito la capacità limitata di richiamare una funzione Lambda. Amazon Cognito può richiamare questa funzione solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `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>"
                }
            }
        }
    ]
}
```

Aggiunta di un trigger Lambda al bacino d'utenza

Come aggiungere un trigger Lambda al bacino d'utenza con la console

1. Usa la console Lambda per creare una funzione Lambda. Per ulteriori informazioni sulle funzioni Lambda, consulta la Guida per gli sviluppatori di AWS Lambda.
2. Passa alla console Amazon Cognito e scegli User Pools (Bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Proprietà del bacino d'utenza e individua Lambda triggers (Trigger Lambda).
5. Scegli Add a Lambda trigger (Aggiungi un trigger Lambda).
6. Seleziona una categoria di trigger Lambda in base alla fase di autenticazione che desideri personalizzare.
7. Seleziona Assign Lambda function (Assegna la funzione Lambda) e seleziona una funzione nella stessa Regione AWS come tuo bacino d'utenza.

   > **Note**
   >
   > Se le tue credenziali AWS Identity and Access Management (IAM) permettono di aggiornare la funzione Lambda, Amazon Cognito aggiunge una policy basata sulle risorse Lambda. Con questa policy, Amazon Cognito può richiamare la funzione selezionata. Se le credenziali di accesso non dispongono di autorizzazioni IAM sufficienti, è necessario aggiornare separatamente la policy basata sulle risorse. Per ulteriori informazioni, consulta the section called “Considerazioni importanti”.

8. Seleziona Save changes (Salva modifiche).
9. Utilizza CloudWatch nella console Lambda per registrare la tua funzione Lambda. Per ulteriori informazioni, consulta Accesso a CloudWatch Logs per Lambda.

Evento trigger Lambda per il bacino d'utenza

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione Lambda restituisce ad Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Questo evento mostra i parametri comuni del trigger Lambda:

JSON

```
{
    "version": "string",
    "triggerSource": "string",
    "region": AWSRegion,
    "userPoolId": "string",
    "userName": "string",
    "callerContext":
        {
            "awsSdkVersion": "string",
            "clientId": "string"
        },
    "request":
        {
            "userAttributes": {
                "string": "string",
                ....
            }
        },
    "response": {}
}
```

Parametri comuni del trigger Lambda del bacino d'utenza

version

Il numero di versione della tua funzione Lambda.

triggerSource

Il nome dell'evento che ha attivato la funzione Lambda. Per una descrizione di ogni triggerSource, consulta Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti.

Regione

Regione AWS come istanza. `AWSRegion`

userPoolId

L'ID del pool di utenti.

userName

Il nome dell'utente corrente.

callerContext

I metadati sulla richiesta e sull'ambiente del codice. Contiene i campi awsSdkVersion e clientId.

awsSdkVersion

La versione dell'SDK AWS che ha generato la richiesta.

clientId

L'ID del client dell'app del pool di utenti.

richiesta

I dettagli della richiesta API dell'utente. Include i seguenti campi e i parametri di richiesta specifici del trigger, Ad esempio, un evento che Amazon Cognito invia a un trigger di pre-autenticazione contiene anche il parametro `userNotFound`. Puoi elaborare il valore di questo parametro per eseguire un'operazione personalizzata quando l'utente tenta di accedere con un nome utente non registrato.

userAttributes

Una o più coppie chiave-valore di nomi e valori di attributi dell'utente, ad esempio `"email": "john@example.com"`.

response

Questo parametro non contiene alcuna informazione nella richiesta originale. La funzione Lambda deve restituire l'intero evento ad Amazon Cognito e aggiungere eventuali parametri di riposta a `response`. Per visualizzare quali parametri di riposta può includere la funzione, fai riferimento alla documentazione relativa al trigger che vuoi usare.

Connessione delle operazioni API ai trigger Lambda

Le sezioni seguenti descrivono i trigger Lambda che Amazon Cognito richiama dall'attività nel pool di utenti.

Quando l'app consente l'accesso agli utenti tramite l'API dei pool di utenti di Amazon Cognito, l'interfaccia utente ospitata o gli endpoint del pool di utenti, Amazon Cognito richiama le funzioni Lambda in base al contesto della sessione. Per ulteriori informazioni sull'API dei pool di utenti di Amazon Cognito e gli endpoint del pool di utenti, consultare Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti. Le tabelle nelle seguenti sezioni descrivono gli eventi che inducono Amazon Cognito a richiamare una funzione e la stringa `triggerSource` che Amazon Cognito include nella richiesta.

Argomenti

- Trigger Lambda nell'API Amazon Cognito
- Trigger Lambda per utenti locali di Amazon Cognito nell'interfaccia utente ospitata
- Trigger Lambda per utenti federati

Trigger Lambda nell'API Amazon Cognito

La tabella seguente descrive le stringhe di origine per i trigger Lambda che possono essere richiamati da Amazon Cognito quando l'app crea, aggiorna o fornisce l'accesso a un utente locale.

Origini dei trigger di utenti locali nell'API Amazon Cognito

Operazione API	Trigger Lambda	Origine del trigger
AdminCreateUser	Preiscrizione	`PreSignUp_AdminCreateUser`
	Pre generazione di token	`TokenGeneration_NewPasswordChallenge`
	Messaggio personalizzato	`CustomMessage_AdminCreateUser`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AdminCreateUser`
	Mittente di SMS personalizzato	`CustomSMSSender_AdminCreateUser`
SignUp	Preiscrizione	`PreSignUp_SignUp`

Operazione API	Trigger Lambda	Origine del trigger
	Messaggio personalizzato	`CustomMessage_SignUp`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_SignUp`
	Mittente di SMS personalizzato	`CustomSMSSender_SignUp`
ConfirmSignUp AdminConfirmSignUp	Post conferma	`PostConfirmation_ConfirmSignUp`
InitiateAuth AdminInitiateAuth	Preautenticazione	`PreAuthentication_Authentication`
	Definizione di una richiesta di autenticazione	`DefineAuthChallenge_Authentication`
	Creazione di una richiesta di autenticazione	`CreateAuthChallenge_Authentication`

	Pre generazione di token	`TokenGeneration_Authentication`
		
		`TokenGeneration_AuthenticateDevice`
		
		`TokenGeneration_RefreshTokens`

	Migrazione degli utenti	`UserMigration_Authentication`
	Messaggio personalizzato	`CustomMessage_Authentication`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AccountTakeOverNotification`
	Mittente di SMS personalizzato	`CustomSMSSender_Authentication`
ForgotPassword	Migrazione degli utenti	`UserMigration_ForgotPassword`
	Messaggio personalizzato	`CustomMessage_ForgotPassword`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_ForgotPassword`
	Mittente di SMS personalizzato	`CustomSMSSender_ForgotPassword`
ConfirmForgotPassword	Post conferma	`PostConfirmation_ConfirmForgotPassword`
UpdateUserAttributes AdminUpdateUserAttributes	Messaggio personalizzato	`CustomMessage_UpdateUserAttribute`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_UpdateUserAttribute`
	Mittente di SMS personalizzato	`CustomSMSSender_UpdateUserAttribute`
VerifyUserAttributes	Messaggio personalizzato	`CustomMessage_VerifyUserAttribute`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_VerifyUserAttribute`

Operazione API	Trigger Lambda	Origine del trigger
	Mittente di SMS personalizzato	`CustomSMSSender_VerifyUserAttribute`

Trigger Lambda per utenti locali di Amazon Cognito nell'interfaccia utente ospitata

La tabella seguente descrive le stringhe di origine per i trigger Lambda che Amazon Cognito può richiamare quando un utente locale accede al pool di utenti con l'interfaccia utente ospitata.

Origini dei trigger dell'utente locale nell'interfaccia utente ospitata

URI dell'interfaccia utente ospitata	Trigger Lambda	Origine del trigger
`/signup`	Preiscrizione	`PreSignUp_SignUp`
	Messaggio personalizzato	`CustomMessage_SignUp`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_SignUp`
	Mittente di SMS personalizzato	`CustomSMSSender_SignUp`
`/confirmuser`	Post conferma	`PostConfirmation_ConfirmSignUp`
`/login`	Preautenticazione	`PreAuthentication_Authentication`
	Definizione di una richiesta di autenticazione	`DefineAuthChallenge_Authentication`
	Creazione di una richiesta di autenticazione	`CreateAuthChallenge_Authentication`
	Pre generazione di token	`TokenGeneration_Authentication`

URI dell'interfaccia utente ospitata	Trigger Lambda	Origine del trigger
		`TokenGeneration_AuthenticateDevice` `TokenGeneration_RefreshTokens`
	Migrazione degli utenti	`UserMigration_Authentication`
	Messaggio personalizzato	`CustomMessage_Authentication`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_AccountTakeOverNotification`
	Mittente di SMS personalizzato	`CustomSMSSender_Authentication`
`/forgotpassword`	Migrazione degli utenti	`UserMigration_ForgotPassword`
	Messaggio personalizzato	`CustomMessage_ForgotPassword`
	Mittente di messaggio e-mail personalizzato	`CustomEmailSender_ForgotPassword`
	Mittente di SMS personalizzato	`CustomSMSSender_ForgotPassword`
`/confirmforgotpassword`	Post conferma	`PostConfirmation_ConfirmForgotPassword`

Trigger Lambda per utenti federati

È possibile utilizzare i seguenti trigger Lambda per personalizzare i flussi di lavoro del bacino d'utenza per gli utenti che accedono con un provider federato.

> **Note**
>
> Gli utenti federati possono accedere utilizzando l'interfaccia utente ospitata di Amazon Cognito oppure puoi generare una richiesta all'Endpoint Authorize che reindirizza gli utenti automaticamente alla pagina di accesso del provider di identità. Non è possibile fornire l'accesso agli utenti federati con l'API del pool di utenti Amazon Cognito.

Origini dei trigger utente federati

Evento di accesso	Trigger Lambda	Origine del trigger
Primo accesso	Preiscrizione	`PreSignUp_ExternalProvider`
	Post conferma	`PostConfirmation_ConfirmSignUp`
	Pre generazione di token	`TokenGeneration_HostedAuth`
Accessi successivi	Preautenticazione	`PreAuthentication_Authentication`
	Post autenticazione	`PostAuthentication_Authentication`
	Pre generazione di token	`TokenGeneration_HostedAuth`

L'accesso federato non richiama Trigger Lambda di richieste di autenticazione personalizzate, Trigger Lambda di migrazione utenti, Trigger Lambda di messaggi personalizzati o Trigger Lambda del mittente personalizzato nel bacino d'utenza.

Connessione dei trigger Lambda alle operazioni funzionali del pool di utenti

Ogni trigger Lambda svolge un ruolo funzionale nel pool di utenti, ad esempio un trigger può modificare il flusso di registrazione o aggiungere una richiesta di autenticazione personalizzata. L'evento inviato da Amazon Cognito a una funzione Lambda può riflettere una delle tante operazioni che costituiscono il ruolo funzionale, ad esempio Amazon Cognito richiama un trigger di pre-registrazione quando un utente si registra e quando viene creato. Ognuno di questi casi diversi per lo stesso ruolo funzionale ha un proprio valore `triggerSource`. La funzione Lambda può elaborare gli eventi in entrata in modo diverso in base all'operazione che l'ha richiamata.

Amazon Cognito richiama tutte le funzioni assegnate anche quando un evento corrisponde a un'origine del trigger, ad esempio quando un utente accede a un pool di utenti a cui hai assegnato i trigger di migrazione utenti e di pre-autenticazione, vengono attivati entrambi i trigger.

Trigger di registrazione, conferma e accesso (autenticazione)

Trigger	Valore triggerSource	Evento
Preiscrizione	`PreSignUp_SignUp`	Pre registrazione.
Preiscrizione	`PreSignUp_AdminCreateUser`	Pre registrazione quando un amministratore crea un nuovo utente.
Preiscrizione	`PreSignUp_ExternalProvider`	Pre-registrazione per provider di identità esterni.
Post conferma	`PostConfirmation_ConfirmSignUp`	Conferma post registrazione.
Post conferma	`PostConfirmation_ConfirmForgotPassword`	Conferma successiva alla password dimenticata.
Preautenticazione	`PreAuthentication_Authentication`	Preautenticazione.
Post autenticazione	`PostAuthentication_Authentication`	Post autenticazione.

Trigger di richieste di autenticazione personalizzate

Trigger	Valore triggerSource	Evento
Definizione di una richiesta di autenticazione	`DefineAuthChallenge_Authentication`	Definizione di una richiesta di autenticazione.
Creazione di una richiesta di autenticazione	`CreateAuthChallenge_Authentication`	Creazione di una richiesta di autenticazione.
Verifica della richiesta di autenticazione	`VerifyAuthChallengeResponse_Authentication`	Verifica di risposta di una richiesta di autenticazione.

Trigger della pre-generazione di token

Trigger	Valore triggerSource	Evento
Pre generazione di token	`TokenGeneration_HostedAuth`	Amazon Cognito autentica l'utente dalla pagina di accesso della tua interfaccia utente ospitata.
Pre generazione di token	`TokenGeneration_Authentication`	Flussi di autenticazione utente completati.
Pre generazione di token	`TokenGeneration_NewPasswordChallenge`	L'amministratore crea l'utente. Amazon Cognito richiama questa operazione quando l'utente deve modificare una password temporanea.
Pre generazione di token	`TokenGeneration_AuthenticateDevice`	Passaggio finale dell'autenticazione di un dispositivo dell'utente.
Pre generazione di token	`TokenGeneration_RefreshTokens`	L'utente cerca di aggiornare l'identità e i token di accesso.

Trigger di migrazione utenti

Trigger	Valore triggerSource	Evento
Migrazione degli utenti	`UserMigration_Authentication`	Migrazione degli utenti al momento dell'accesso.
Migrazione degli utenti	`UserMigration_ForgotPassword`	Migrazione degli utenti durante il flusso di password dimenticata.

Trigger di messaggi personalizzati

Trigger	Valore triggerSource	Evento
Messaggio personalizzato	`CustomMessage_SignUp`	Messaggio personalizzato quando un utente effettua la registrazione nel bacino d'utenza.
Messaggio personalizzato	`CustomMessage_AdminCreateUser`	Messaggio personalizzato quando crei un utente come amministratore e Amazon Cognito gli invia una password temporanea.
Messaggio personalizzato	`CustomMessage_ResendCode`	Messaggio personalizzato quando un utente esistente richiede un nuovo codice di conferma.
Messaggio personalizzato	`CustomMessage_ForgotPassword`	Messaggio personalizzato quando l'utente richiede la reimpostazione della password.
Messaggio personalizzato	`CustomMessage_UpdateUserAttribute`	Messaggio personalizzato quando un utente modifica il proprio indirizzo e-mail o

Trigger	Valore triggerSource	Evento
		numero di telefono e Amazon Cognito gli invia un codice di verifica.
Messaggio personalizzato	`CustomMessage_VerifyUserAttribute`	Messaggio personalizzato quando un utente aggiunge un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice di verifica.
Messaggio personalizzato	`CustomMessage_Authentication`	Messaggio personalizzato quando un utente che ha configurato SMS MFA esegue l'accesso.

Trigger Lambda di pre-registrazione

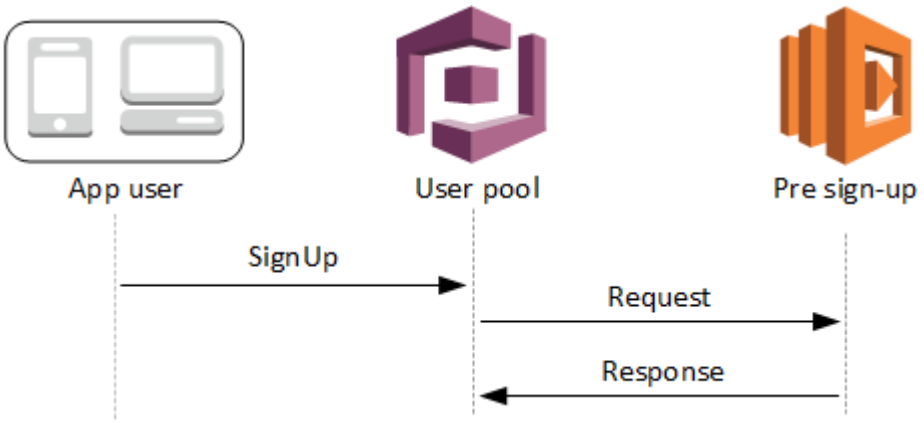
Poco prima della registrazione di un nuovo utente in Amazon Cognito, viene attivata la funzione di pre-registrazione AWS Lambda. Puoi utilizzare questa funzione nell'ambito del processo di registrazione per eseguire una convalida personalizzata e, in base ai risultati della convalida, accettare o rifiutare la richiesta di registrazione.

Argomenti

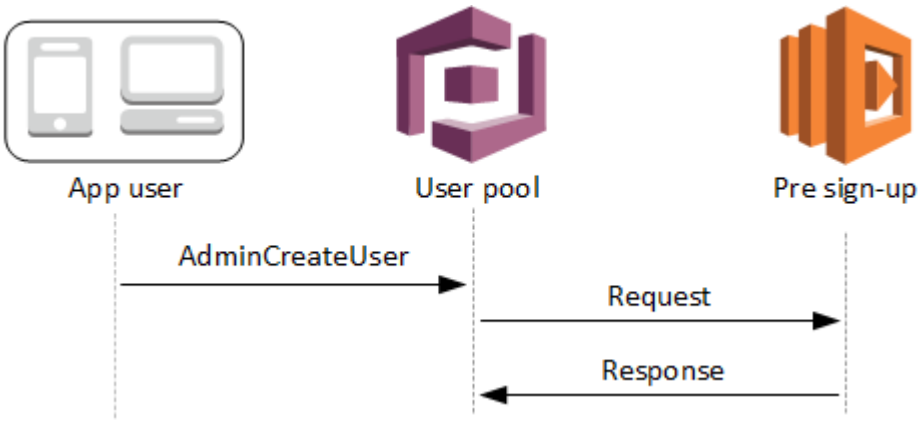
- Flussi Lambda di pre-registrazione
- Parametri del trigger Lambda di pre-registrazione
- Tutorial sulla registrazione
- Esempio di pre-registrazione: conferma automatica degli utenti di un dominio registrato
- Esempio di pre-registrazione: conferma e verifica automatica di tutti gli utenti
- Esempio di pre-registrazione: registrazione rifiutata se il nome utente ha meno di cinque caratteri

Flussi Lambda di pre-registrazione

Flusso di registrazione client



Flusso di registrazione server



La richiesta include i dati di convalida dal client. Questi dati provengono dai `ValidationData` valori passati al pool di utenti SignUp e ai metodi AdminCreateUser API.

Parametri del trigger Lambda di pre-registrazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
```

```
        },
        "validationData": {
            "string": "string",
            . . .
         },
        "clientMetadata": {
            "string": "string",
            . . .
         }
    },

    "response": {
        "autoConfirmUser": "boolean",
        "autoVerifyPhone": "boolean",
        "autoVerifyEmail": "boolean"
    }
}
```

Parametri di richiesta di pre registrazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente. I nomi di attributo sono le chiavi.

validationData

Una o più coppie chiave-valore con i dati degli attributi utente che la tua app ha passato ad Amazon Cognito nella richiesta di creazione di un nuovo utente. Invia queste informazioni alla tua funzione Lambda nel ValidationData parametro della tua richiesta AdminCreateUsero SignUpAPI.

Amazon Cognito non imposta ValidationData i tuoi dati come attributi dell'utente che crei. ValidationData sono informazioni temporanee sull'utente fornite ai fini del trigger Lambda prima della registrazione.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di pre-registrazione. Puoi passare questi dati alla tua funzione Lambda utilizzando il ClientMetadata parametro nelle seguenti azioni API: AdminCreateUser, AdminRespondToAuthChallengeForgotPassword, e. SignUp

Parametri di risposta di pre-registrazione

Nella risposta, puoi impostare `autoConfirmUser` su `true` se desideri auto confermare l'utente. Puoi impostare `autoVerifyEmail` su `true` per auto verificare l'e-mail dell'utente. Puoi impostare `autoVerifyPhone` su `true` per auto verificare il numero di telefono dell'utente.

> **Note**
>
> I parametri di risposta `autoVerifyPhone`, `autoVerifyEmail` e `autoConfirmUser` vengono ignorati da Amazon Cognito quando la funzione Lambda di pre-registrazione viene attivata dall'API `AdminCreateUser`.

autoConfirmUser

Impostare su `true` per auto confermare l'utente o altrimenti su `false`.

autoVerifyEmail

Imposta su `true` per definire come verificato il messaggio e-mail di un utente che sta effettuando la registrazione o altrimenti su `false`. Se `autoVerifyEmail` è impostato su `true`, l'attributo della `email` deve avere un valore valido e non nullo. In caso contrario, si verificherà un errore e l'utente non sarà in grado di completare la registrazione.

Se l'attributo `email` è selezionato come alias, viene creato un alias per l'indirizzo e-mail dell'utente quando è impostato `autoVerifyEmail`. Se esiste già un alias con quell'indirizzo e-mail, l'alias verrà spostato al nuovo utente e l'indirizzo e-mail dell'utente precedente verrà contrassegnato come non verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

autoVerifyPhone

Imposta su `true` per impostare come verificato il numero di telefono di un utente che sta effettuando la registrazione o altrimenti su `false`. Se `autoVerifyPhone` è impostato su `true`, l'attributo della `phone_number` deve avere un valore valido e non nullo. In caso contrario, si verificherà un errore e l'utente non sarà in grado di completare la registrazione.

Se l'attributo del `phone_number` è selezionato come un alias, un alias verrà creato per il numero di telefono dell'utente quando `autoVerifyPhone` è impostato. Se esiste già un alias con quel numero di telefono, l'alias verrà spostato al nuovo utente e il numero di telefono dell'utente

precedente verrà contrassegnato come non verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Tutorial sulla registrazione

La funzione Lambda di pre-registrazione viene attivata prima della registrazione dell'utente. Guarda questi tutorial di iscrizione ad Amazon Cognito per JavaScript Android e iOS.

Piattaforma	Tutorial
JavaScript SDK per le identità	Registra gli utenti con JavaScript
SDK di identità per Android	Registrazione degli utenti con Android
SDK di identità per iOS	Registrazione degli utenti con iOS

Esempio di pre-registrazione: conferma automatica degli utenti di un dominio registrato

Puoi utilizzare il trigger Lambda di pre-registrazione per aggiungere una logica personalizzata che convalida i nuovi utenti che si registrano al tuo bacino d'utenza. Questo è un JavaScript programma di esempio che mostra come registrare un nuovo utente. Richiama un trigger Lambda di pre-registrazione nell'ambito del processo di autenticazione.

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());
  }
);
```

Questo è un trigger Lambda di esempio richiamato appena prima della registrazione con il trigger Lambda di pre-registrazione del bacino d'utenza. Utilizza un attributo personalizzato custom:domain per confermare automaticamente i nuovi utenti di un determinato dominio e-mail. Tutti i nuovi utenti che non fanno parte del dominio personalizzato verranno aggiunti al bacino d'utenza, ma non saranno confermati automaticamente.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "request": {
```

```
        "userAttributes": {
            "email": "testuser@example.com",
            "custom:domain": "example.com"
        }
    },
    "response": {}
}
```

Esempio di pre-registrazione: conferma e verifica automatica di tutti gli utenti

Questo esempio conferma tutti gli utenti e imposta lo stato di avvenuta verifica per gli attributi `email` e `phone_number` dell'utente se presenti. Inoltre, se l'aliasing è abilitato, saranno creati gli alias per `phone_number` e `email` nei casi in cui è impostata la verifica automatica.

> **Note**
>
> Se esiste già un alias con lo stesso numero di telefono, l'alias verrà spostato al nuovo utente e il `phone_number` dell'utente precedente verrà contrassegnato come non verificato. Lo stesso vale per gli indirizzi e-mail. Per evitare che ciò accada, puoi utilizzare l'ListUsers API dei pool di utenti per vedere se esiste un utente esistente che sta già utilizzando il numero di telefono o l'indirizzo e-mail del nuovo utente come 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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "request": {
    "userAttributes": {
      "email": "user@example.com",
      "phone_number": "+12065550100"
    }
  },
  "response": {}
}
```

Esempio di pre-registrazione: registrazione rifiutata se il nome utente ha meno di cinque caratteri

Questo esempio verifica la lunghezza del nome utente in una richiesta di registrazione. L'esempio restituisce un errore se l'utente immette un nome con meno di cinque caratteri.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "userName": "rroe",
  "response": {}
```

```
}
```

Trigger Lambda di post-conferma

Amazon Cognito richiama questo trigger dopo che un utente registrato conferma il proprio account utente. Nella funzione Lambda di post conferma, puoi inviare messaggi personalizzati o aggiungere richieste API personalizzate. Ad esempio, puoi eseguire query su un sistema esterno e compilare attributi aggiuntivi per l'utente. Amazon Cognito richiama questo trigger solo per gli utenti che effettuano la registrazione al pool di utenti, non per gli account utente creati con le tue credenziali dell'amministratore.

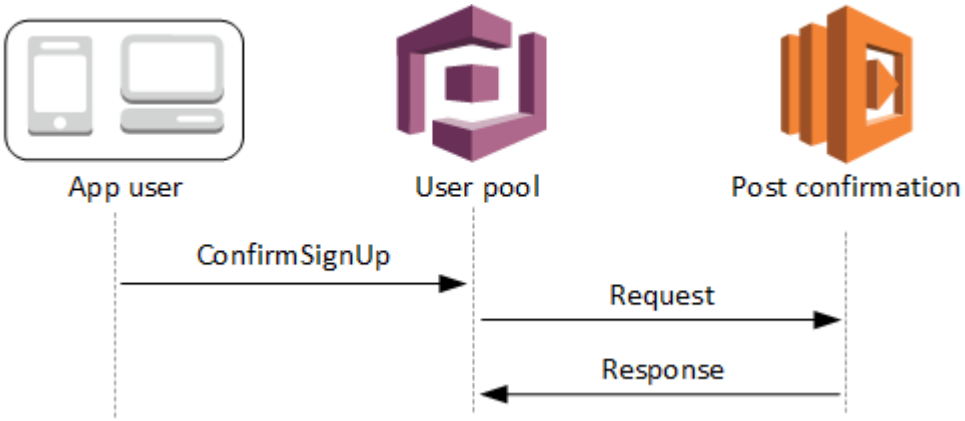
La richiesta contiene gli attributi attuali per l'utente confermato.

Argomenti

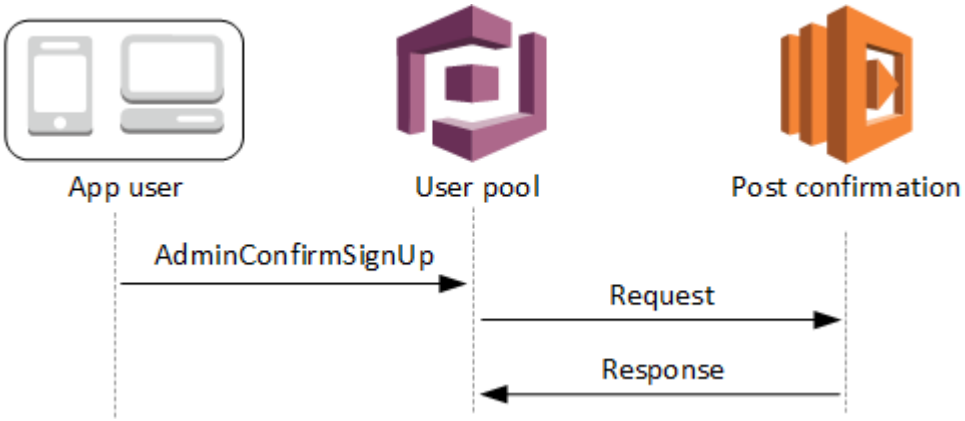
- Flussi Lambda di post-conferma
- Parametri del trigger Lambda di post-conferma
- Tutorial sulla conferma degli utenti
- Esempio di post conferma

Flussi Lambda di post-conferma

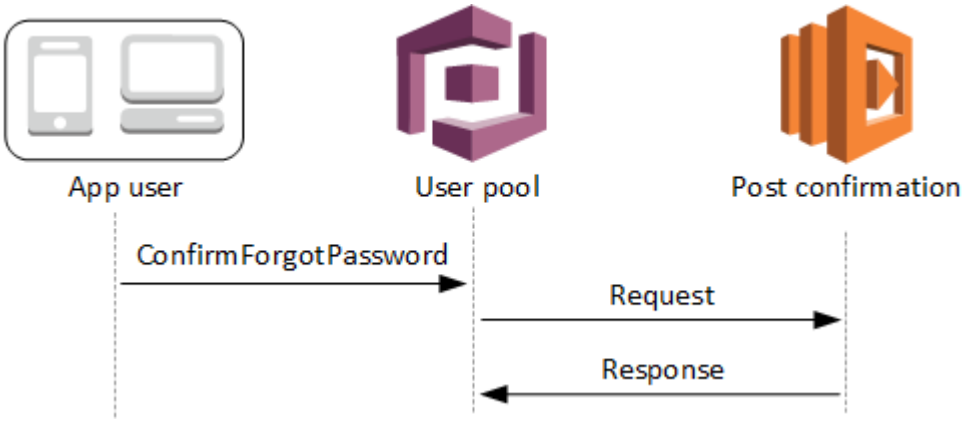
Flusso di conferma registrazione client



Flusso di conferma registrazione server



Flusso di conferma password dimenticata



Parametri del trigger Lambda di post-conferma

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
            "userAttributes": {
                "string": "string",
                . . .
            },
            "clientMetadata": {
             "string": "string",
              . . .
            }
```

```
        },
    "response": {}
}
```

Parametri di richiesta di post-conferma

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di post-conferma. È possibile passare questi dati alla funzione Lambda utilizzando il parametro ClientMetadata nelle seguenti operazioni API: AdminConfirmSignUp, ConfirmForgotPassword, ConfirmSignUp e SignUp.

Parametri di risposta post-conferma

Nella risposta non è prevista la restituzione di alcuna informazione aggiuntiva.

Tutorial sulla conferma degli utenti

La funzione Lambda di post-conferma viene attivata appena dopo l'avvenuta conferma da parte di Amazon Cognito di un nuovo utente. Guarda questi tutorial sulla conferma degli utenti per JavaScript, Android e iOS.

Piattaforma	Tutorial
SDK di identità per JavaScript	Conferma degli utenti con JavaScript
SDK di identità per Android	Conferma degli utenti con Android
SDK di identità per iOS	Conferma degli utenti con iOS

Esempio di post conferma

Questa funzione Lambda di esempio invia un messaggio e-mail di conferma all'utente utilizzando Amazon SES. Per ulteriori informazioni, consulta la Guida per gli sviluppatori di 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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "request": {
        "userAttributes": {
            "email": "user@example.com",
            "email_verified": true
        }
    },
    "response": {}
}
```

Trigger Lambda di pre-autenticazione

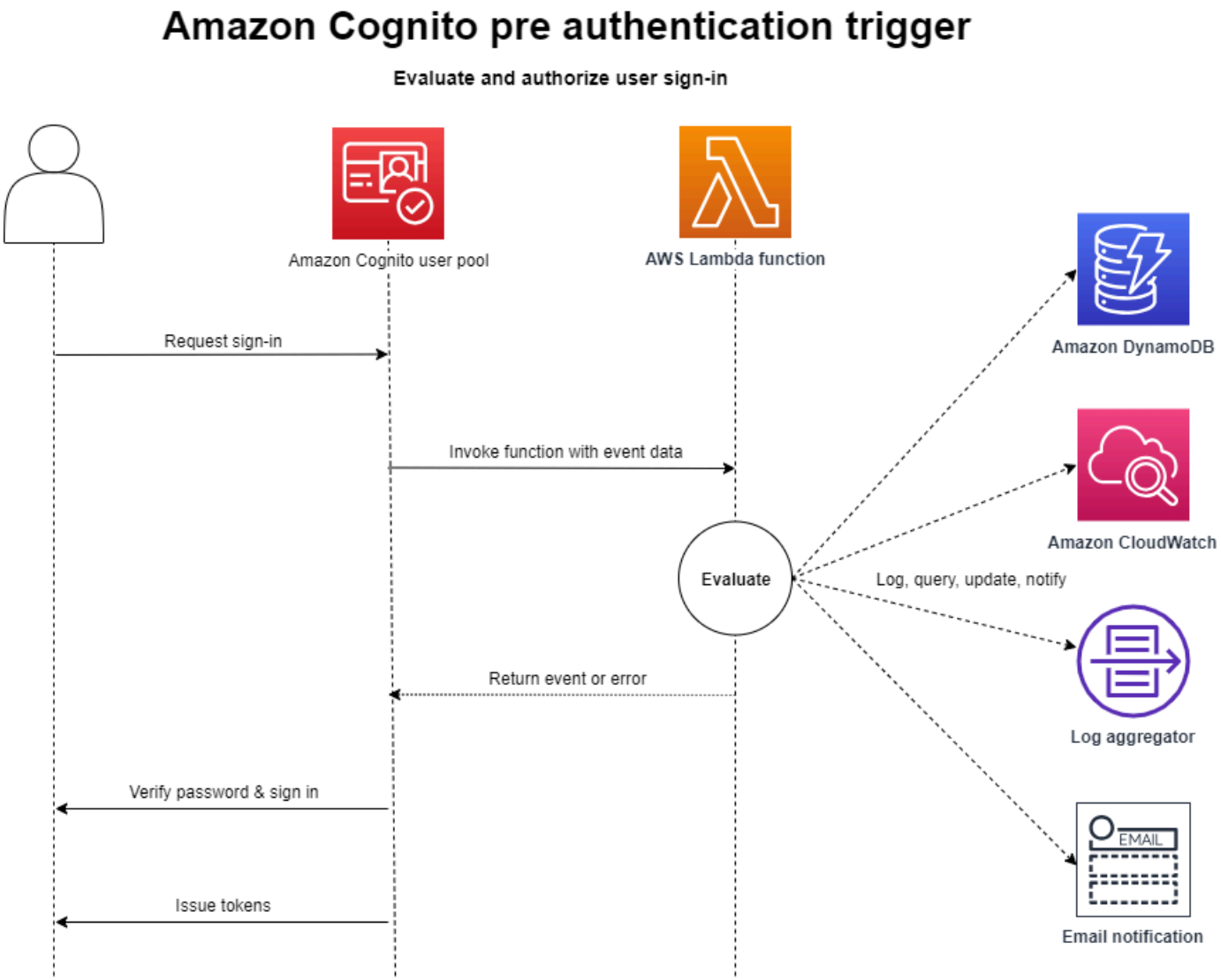
Amazon Cognito richiama questo trigger quando un utente prova a effettuare l'accesso, in modo da poter creare la convalida personalizzata che esegue azioni preparatorie. Ad esempio, puoi rifiutare la richiesta di autenticazione o registrare i dati di sessione in un sistema esterno.

> **Note**
>
> Questo trigger Lambda non si attiva quando un utente non esiste o dispone già di una sessione esistente nel pool di utenti. Se l'impostazione `PreventUserExistenceErrors` di un client di app per pool di utenti è impostata su `ENABLED`, il trigger Lambda si attiverà.

Argomenti

- Panoramica sul flusso dell'autenticazione
- Parametri di attivazione Lambda di pre-autenticazione
- Esempio di pre autenticazione

Panoramica sul flusso dell'autenticazione



La richiesta include i dati di convalida dal client che provengono dai valori di `ClientMetadata` trasmessi dall'app alle operazioni API `InitiateAuth` e `AdminInitiateAuth` del pool di utenti.

Per ulteriori informazioni, consulta Flusso di autenticazione del bacino d'utenza.

Parametri di attivazione Lambda di pre-autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "validationData": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {}
}
```

Parametri di richiesta di pre-autenticazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

userNotFound

Quando imposti `PreventUserExistenceErrors` su `ENABLED` per il tuo client del bacino d'utenza, Amazon Cognito popola questo valore booleano.

validationData

Una o più coppie chiave-valore contenenti i dati di convalida nella richiesta di registrazione dell'utente. Per trasmettere questi dati alla funzione Lambda utilizza il parametro ClientMetadata nelle operazioni API InitiateAuth e AdminInitiateAuth.

Parametri di risposta di pre-autenticazione

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta. La funzione può restituire un errore per rifiutare il tentativo di accesso o utilizzare le operazioni API per eseguire query e modificare le risorse.

Esempio di pre autenticazione

Questa funzione di esempio impedisce agli utenti di effettuare l'accesso al pool di utenti con un client dell'app specifico. Poiché la funzione Lambda di preautenticazione non viene richiamata quando l'utente dispone di una sessione esistente, questa funzione impedisce solo nuove sessioni con l'ID del client dell'app che desideri bloccare.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
    "callerContext": {
        "clientId": "<user pool app client id to be blocked>"
    },
    "response": {}
}
```

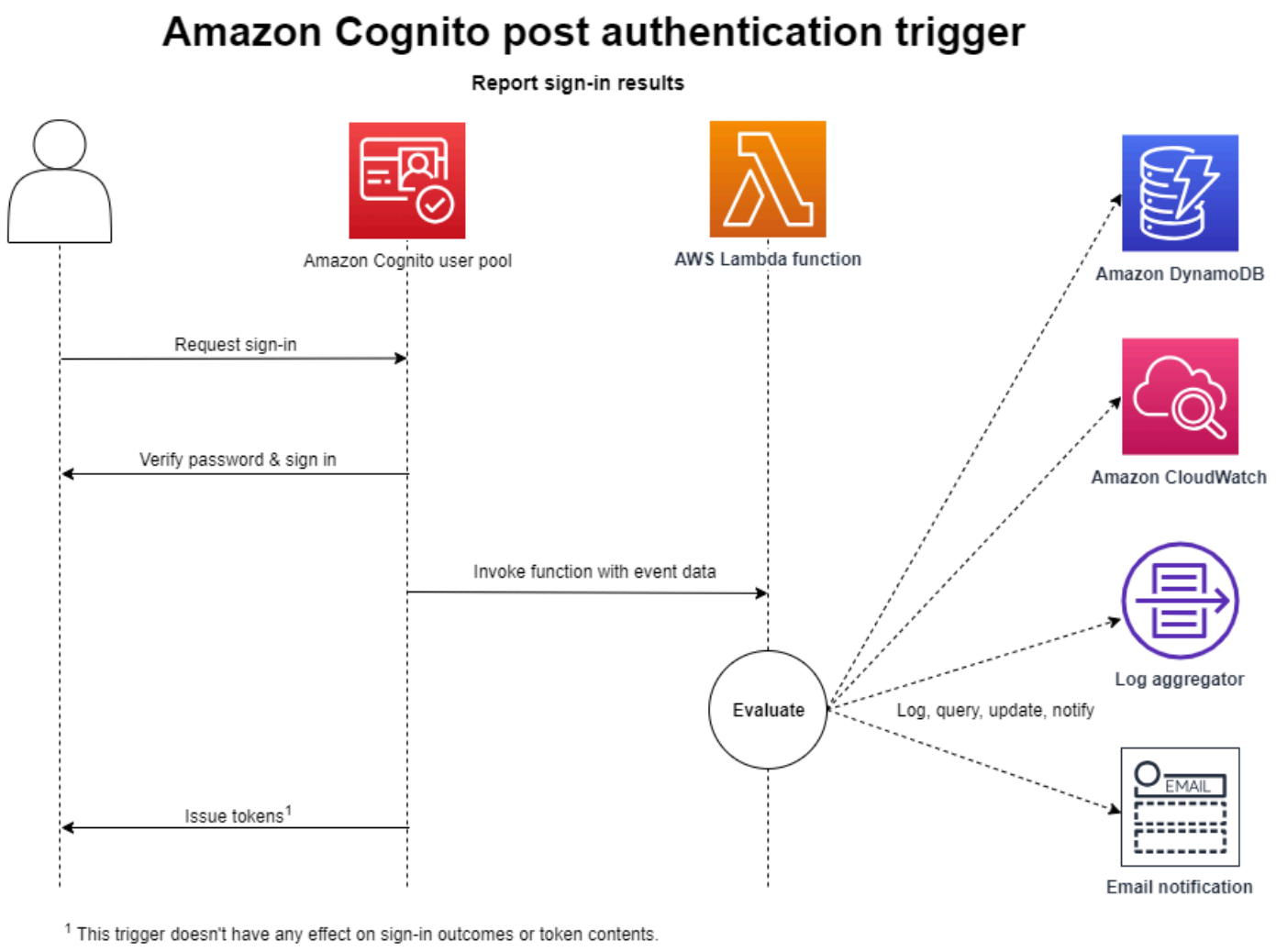
Trigger Lambda di post-autenticazione

Amazon Cognito richiama questo trigger dopo l'accesso di un utente, il che ti consente di aggiungere logica personalizzata dopo l'autenticazione da parte di Amazon Cognito.

Argomenti

- Panoramica sul flusso dell'autenticazione
- Parametri del trigger Lambda di post autenticazione
- Tutorial sull'autenticazione
- Esempio di post autenticazione

Panoramica sul flusso dell'autenticazione



Per ulteriori informazioni, consulta Flusso di autenticazione del bacino d'utenza.

Parametri del trigger Lambda di post autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
```

```
            . . .
        },
        "newDeviceUsed": boolean,
        "clientMetadata": {
            "string": "string",
            . . .
           }
       },
    "response": {}
}
```

Parametri di richiesta di post autenticazione

newDeviceUsed

Questo contrassegno indica se l'utente ha effettuato l'accesso da un nuovo dispositivo. Amazon Cognito imposta questo contrassegno solo se il valore dei dispositivi memorizzati del bacino d'utenza è impostato su `Always` o `User Opt-In`.

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di post-autenticazione. Per trasmettere questi dati alla funzione Lambda, usa il parametro ClientMetadata nelle operazioni API AdminRespondToAuthChallenge e RespondToAuthChallenge. Quando trasmette la richiesta alla funzione di post autenticazione, Amazon Cognito non include dati provenienti dal parametro ClientMetadata nelle operazioni API AdminInitiateAuth e InitiateAuth.

Parametri di risposta di post autenticazione

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta. La funzione può utilizzare le operazioni API per interrogare e modificare le risorse o registrare i metadati degli eventi in un sistema esterno.

Tutorial sull'autenticazione

L'approvazione dell'accesso di un utente da parte di Amazon Cognito attiva la funzione Lambda di post-autenticazione. Guarda questi tutorial sulla procedura di accesso per JavaScript, Android e iOS.

Piattaforma	Tutorial
SDK di identità per JavaScript	Accesso degli utenti con JavaScript
SDK di identità per Android	Accesso degli utenti con Android
SDK di identità per iOS	Accesso degli utenti con iOS

Esempio di post autenticazione

Questa funzione Lambda di post-autenticazione di esempio invia i dati di un accesso riuscito a CloudWatch Logs.

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'])
```

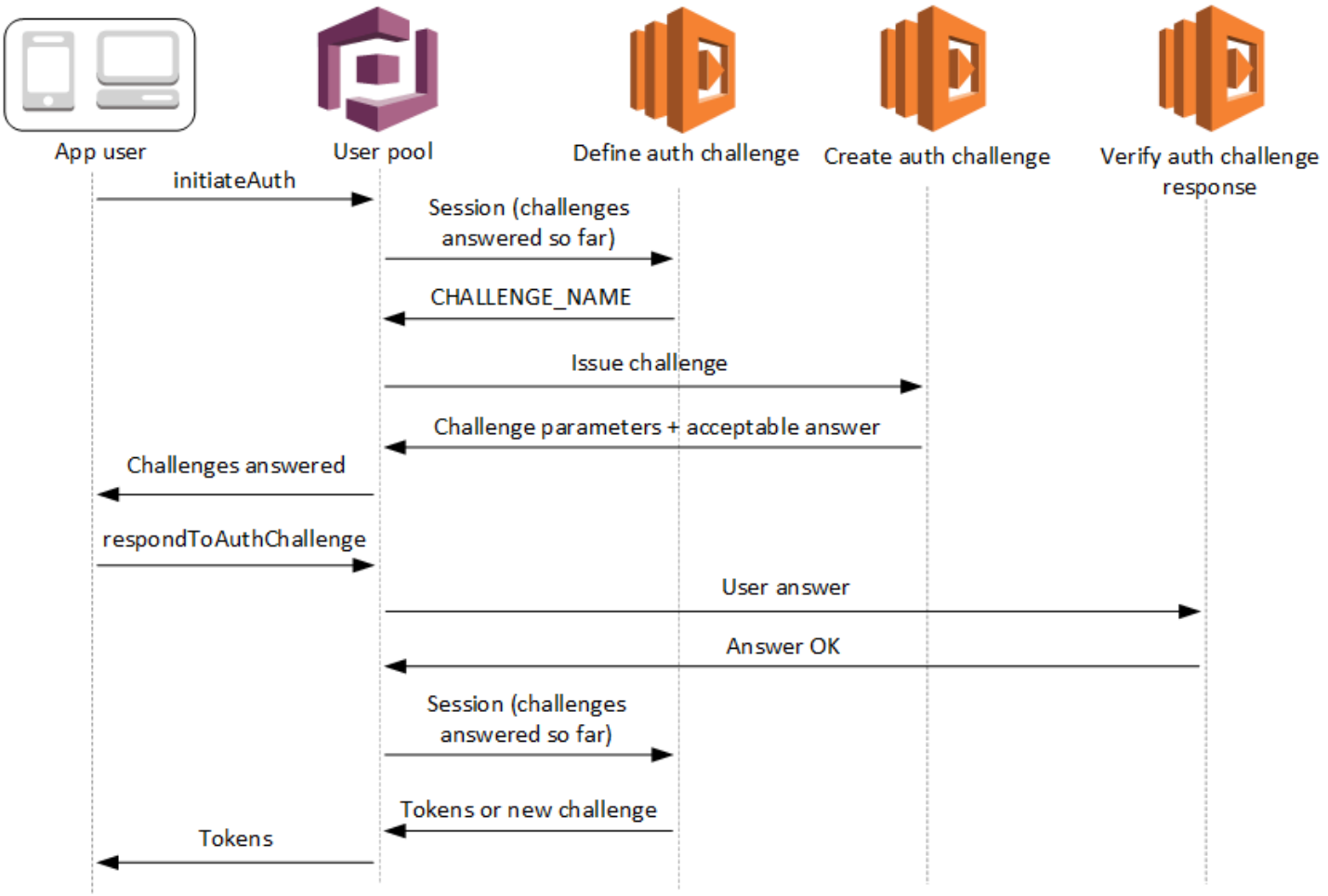
```
    print ("User ID = ", event['userName'])

    # Return to Amazon Cognito
    return event
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "triggerSource": "testTrigger",
  "userPoolId": "testPool",
  "userName": "testName",
  "callerContext": {
      "clientId": "12345"
  },
  "response": {}
}
```

Trigger Lambda di richieste di autenticazione personalizzate



Questi trigger Lambda generano e verificano le proprie richieste come parte del flusso di autenticazione personalizzato di un bacino d'utenza.

Definizione di una richiesta di autenticazione

Amazon Cognito richiama questo trigger per avviare il flusso di autenticazione personalizzato.

Creazione di una richiesta di autenticazione

Amazon Cognito richiama questo trigger dopo la definizione di una richiesta di autenticazione per creare una richiesta personalizzata.

Verifica della risposta a una richiesta di autenticazione

Amazon Cognito richiama questo trigger per verificare se la risposta dell'utente finale per una richiesta personalizzata è valida o meno.

Con questi trigger Lambda di richiesta è possibile integrare nuovi tipi di richieste. Ad esempio, questi tipi di richiesta potrebbero includere domande dinamiche o CAPTCHA.

È possibile generalizzare l'autenticazione in due fasi comuni utilizzando i metodi API InitiateAuth e RespondToAuthChallenge del bacino d'utenza.

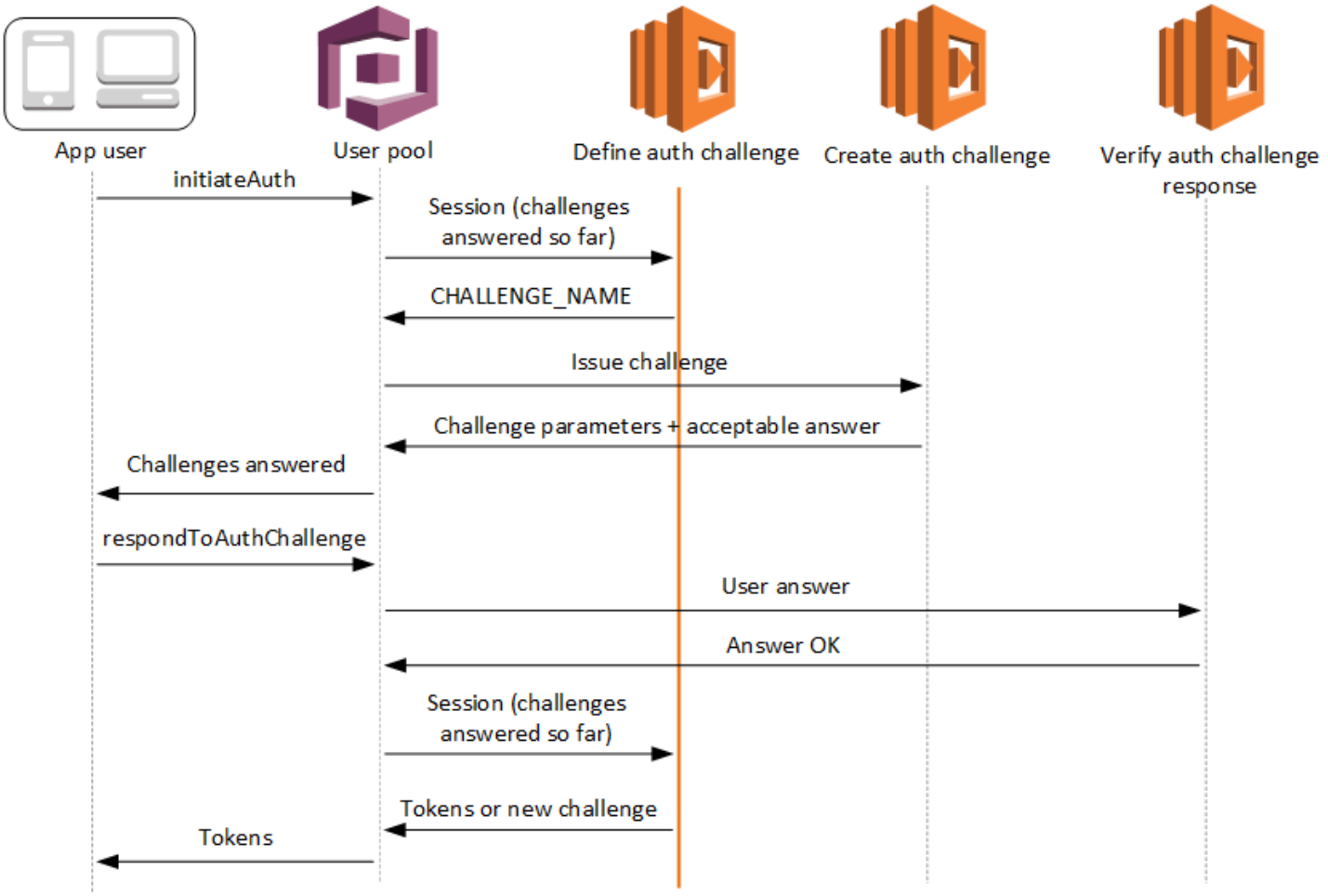
In questo flusso, l'autenticazione di un utente viene effettuata rispondendo a richieste successive di autenticazione finché l'utente non viene bloccato in caso di esito negativo oppure riceve i relativi token. Queste due chiamate API possono essere ripetute per includere diverse richieste.

> **Note**
>
> L'interfaccia utente ospitata di Amazon Cognito non supporta l'autenticazione personalizzata con trigger Lambda di richiesta di autenticazione personalizzati.

Argomenti

- Definizione del trigger Lambda di una richiesta di autenticazione
- Creazione del trigger Lambda di una richiesta di autenticazione
- Trigger Lambda di una verifica di risposta di una richiesta di autenticazione

Definizione del trigger Lambda di una richiesta di autenticazione



Definizione di una richiesta di autenticazione

Amazon Cognito richiama questo trigger per avviare il [flusso di autenticazione personalizzato](#).

La richiesta per questo trigger Lambda include `session`. Il parametro `session` è una matrice contenente tutte le sfide a cui l'utente viene sottoposto nel processo di autenticazione in corso. La richiesta include anche il risultato corrispondente. La matrice `session` archivia i dettagli della sfida (`ChallengeResult`) in ordine cronologico. La richiesta `session[0]` rappresenta la prima sfida che l'utente riceve.

Puoi fare in modo che Amazon Cognito verifichi le password degli utenti prima di emettere le tue richieste personalizzate. Tutti i trigger Lambda associati alla categoria Autenticazione delle [quote del tasso di richiesta](#) verranno eseguiti quando si esegue l'autenticazione SRP in un flusso di richiesta personalizzato. Ecco una panoramica del processo:

1. La tua app avvia la procedura di accesso chiamando `InitiateAuth` o `AdminInitiateAuth` con la mappatura `AuthParameters`. I parametri devono includere `CHALLENGE_NAME: SRP_A,` e i valori per `SRP_A` e `USERNAME`.
2. Amazon Cognito invoca il tuo trigger Lambda define auth challenge con una sessione iniziale contenente `challengeName: SRP_A` e `challengeResult: true`.
3. Dopo aver ricevuto questi input, la funzione Lambda risponde con `challengeName: PASSWORD_VERIFIER`, `issueTokens: false`, `failAuthentication: false`.
4. Se la verifica della password ha esito positivo, Amazon Cognito richiama nuovamente la funzione Lambda con una nuova sessione contenente `challengeName: PASSWORD_VERIFIER` e `challengeResult: true`.
5. La funzione Lambda avvia le tue sifde personalizzate rispondendo con `challengeName: CUSTOM_CHALLENGE`, `issueTokens: false` e `failAuthentication: false`. Se non desideri avviare il flusso di autenticazione personalizzato con la verifica della password, è possibile avviare l'accesso con la mappa `AuthParameters` che include `CHALLENGE_NAME: CUSTOM_CHALLENGE`.
6. Il loop di sfide si ripete finché non viene data risposta a tutte le sfide.

Argomenti

- Parametri del trigger Lambda di definizione delle richieste di autenticazione
- Definizione di un esempio di una richiesta di autenticazione

Parametri del trigger Lambda di definizione delle richieste di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
                . . .
        },
        "session": [
            ChallengeResult,
            . . .
```

```
        ],
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "challengeName": "string",
        "issueTokens": boolean,
        "failAuthentication": boolean
    }
}
```

Parametri di richiesta di definizione sfida di autenticazione

Quando Amazon Cognito richiama la tua funzione Lambda, fornisce i seguenti parametri:

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

userNotFound

Valore booleano che viene popolato da Amazon Cognito quando `PreventUserExistenceErrors` è impostato su `ENABLED` per il client del bacino d'utenza. Un valore `true` indica che l'ID utente (nome utente, indirizzo e-mail e altri dettagli) non corrisponde ad alcun utente esistente. Quando `PreventUserExistenceErrors` è impostato su `ENABLED`, il servizio non informa l'app riguardo a utenti inesistenti. Raccomandiamo che le tue funzioni Lambda mantengano la stessa esperienza utente e tengano conto della latenza. In questo modo, l'intermediario non può rilevare comportamenti diversi quando l'utente esiste o non esiste.

session

Una matrice di elementi `ChallengeResult`. Ognuna contiene i seguenti elementi:

challengeName

Uno dei seguenti tipi di richieste: `CUSTOM_CHALLENGE`, `SRP_A`, `PASSWORD_VERIFIER`, `SMS_MFA, DEVICE_SRP_AUTH`, `DEVICE_PASSWORD_VERIFIER` o `ADMIN_NO_SRP_AUTH`.

Quando la funzione Definisci la sfida di autenticazione genera una sfida `PASSWORD_VERIFIER` per un utente che ha impostato l'autenticazione a più fattori, viene

completata da Amazon Cognito con una sfida `SMS_MFA`. Nella funzione, includi la gestione degli eventi di input da sfide `SMS_MFA`. Non è necessario richiamare la sfida `SMS_MFA` dalla funzione Definisci la sfida di autenticazione.

> **Important**
>
> Quando la funzione determina se un utente ha completato l'autenticazione ed è necessario emettere token, controlla sempre `challengeName` nella funzione Definisci la sfida di autenticazione e verifica che corrisponda al valore previsto.

challengeResult

Imposta su `true` se l'utente ha completato con successo la sfida o altrimenti su `false`.

challengeMetaData

Il tuo nome per la sfida personalizzata. Usato solo se `challengeName` è `CUSTOM_CHALLENGE`.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di definizione di una richiesta di autenticazione. Puoi trasmettere questi dati alla funzione Lambda utilizzando il parametro `ClientMetadata` nelle operazioni API AdminRespondToAuthChallenge e RespondToAuthChallenge. La richiesta che richiama la funzione define auth challenge non include i dati trasmessi al parametro ClientMetadata nelle operazioni API AdminInitiateAuth e InitiateAuth.

Parametri di risposta di definizione sfida di autenticazione

Nella risposta, puoi restituire la prossima fase del processo di autenticazione.

challengeName

Una stringa che include il nome della sfida successiva. Se desideri presentare una nuova sfida al tuo utente, specifica qui il nome della sfida.

issueTokens

Se stabilisci che l'utente abbia completato abbastanza sfide per essere autenticato, imposta su `true`. Se l'utente non ha soddisfatto i criteri delle sfide, imposta su `false`.

failAuthentication

Se desideri terminare l'attuale processo di autenticazione, imposta su `true`. Per continuare il processo di autenticazione corrente, imposta su `false`.

Definizione di un esempio di una richiesta di autenticazione

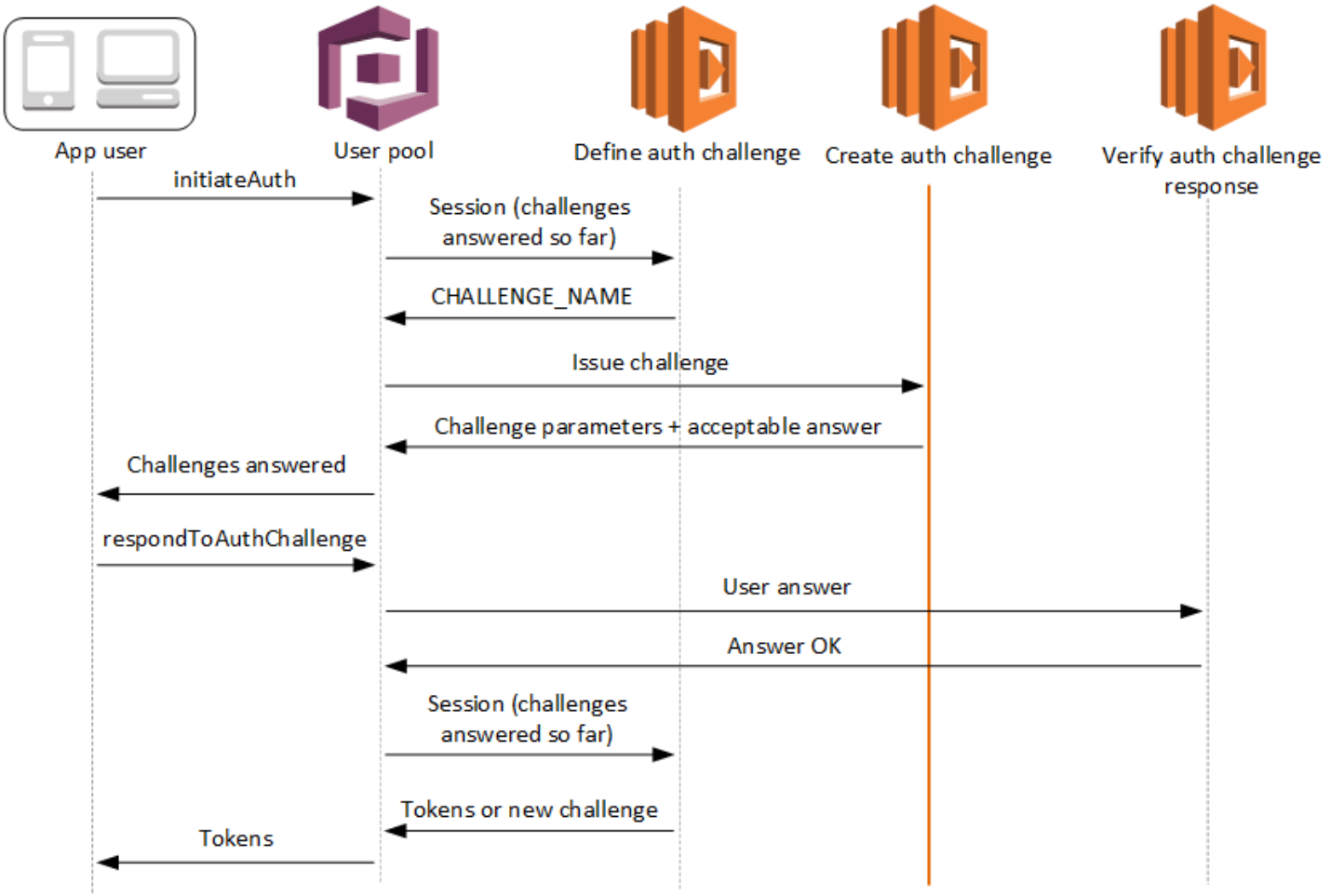
Questo esempio definisce una serie di sfide per l'autenticazione e rilascia token solo se l'utente completa correttamente tutte le sfide.

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 }
```

Creazione del trigger Lambda di una richiesta di autenticazione



Creazione di una richiesta di autenticazione

Amazon Cognito richiama questo trigger dopo la definizione di una richiesta di autenticazione nel caso in cui una richiesta personalizzata sia stata specificata come parte del trigger di definizione di una richiesta di autenticazione. Crea un flusso di autenticazione personalizzato.

Questo trigger Lambda viene richiamato per creare una richiesta da presentare all'utente. La richiesta per questo trigger Lambda include `challengeName` e `session`. Il `challengeName` è una stringa, nonché il nome della prossima sfida per l'utente. Il valore di questo attributo è impostato nel trigger Lambda di definizione di una richiesta di autenticazione.

Il loop di sfide si ripeterà finché non viene data risposta a tutte le sfide.

Argomenti

- Parametri del trigger Lambda di creazione di una richiesta di autenticazione
- Creazione di un esempio di una richiesta di autenticazione

Parametri del trigger Lambda di creazione di una richiesta di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "challengeName": "string",
        "session": [
            ChallengeResult,
            . . .
        ],
        "clientMetadata": {
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "publicChallengeParameters": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
```

```
            "string": "string",
            . . .
        },
        "challengeMetadata": "string"
    }
}
```

Parametri di richiesta di creazione sfida di autenticazione

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

userNotFound

Questo valore booleano viene popolato quando `PreventUserExistenceErrors` è impostato su `ENABLED` per il client del bacino d'utenza.

challengeName

Il nome della nuova sfida.

session

L'elemento della sessione è una matrice di elementi `ChallengeResult`, ognuno dei quali contiene i seguenti elementi:

challengeName

Il tipo di sfida. Uno tra: `"CUSTOM_CHALLENGE"`, `"PASSWORD_VERIFIER"`, `"SMS_MFA"`, `"DEVICE_SRP_AUTH"`, `"DEVICE_PASSWORD_VERIFIER"`, o `"ADMIN_NO_SRP_AUTH"`.

challengeResult

Imposta su `true` se l'utente ha completato con successo la sfida o altrimenti su `false`.

challengeMetaData

Il tuo nome per la sfida personalizzata. Usato solo se `challengeName` è `"CUSTOM_CHALLENGE"`.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di creazione di una richiesta di autenticazione. Puoi trasmettere questi dati alla funzione Lambda utilizzando il parametro ClientMetadata nelle operazioni API

AdminRespondToAuthChallenge e RespondToAuthChallenge. La richiesta che richiama la funzione create auth challenge non include i dati trasmessi al parametro ClientMetadata nelle operazioni API AdminInitiateAuth e InitiateAuth.

Parametri di risposta di creazione sfida di autenticazione

publicChallengeParameters

Una o più coppie chiave-valore per l'app del client da usare nella sfida che deve essere presentata all'utente. Questo parametro deve contenere tutte le informazioni necessarie per presentare, in maniera accurata, la sfida all'utente.

publicChallengeParameters

Questo parametro viene utilizzato solo dal trigger Lambda di una verifica di risposta di una richiesta di autenticazione. Questo parametro dovrebbe contenere tutte le informazioni necessarie per convalidare la risposta dell'utente alla sfida. In altre parole, il parametro `publicChallengeParameters` contiene la domanda che viene presentata all'utente e `privateChallengeParameters` contiene le risposte valide per la domanda.

challengeMetaData

Il tuo nome per la sfida personalizzata, se questa è una sfida personalizzata.

Creazione di un esempio di una richiesta di autenticazione

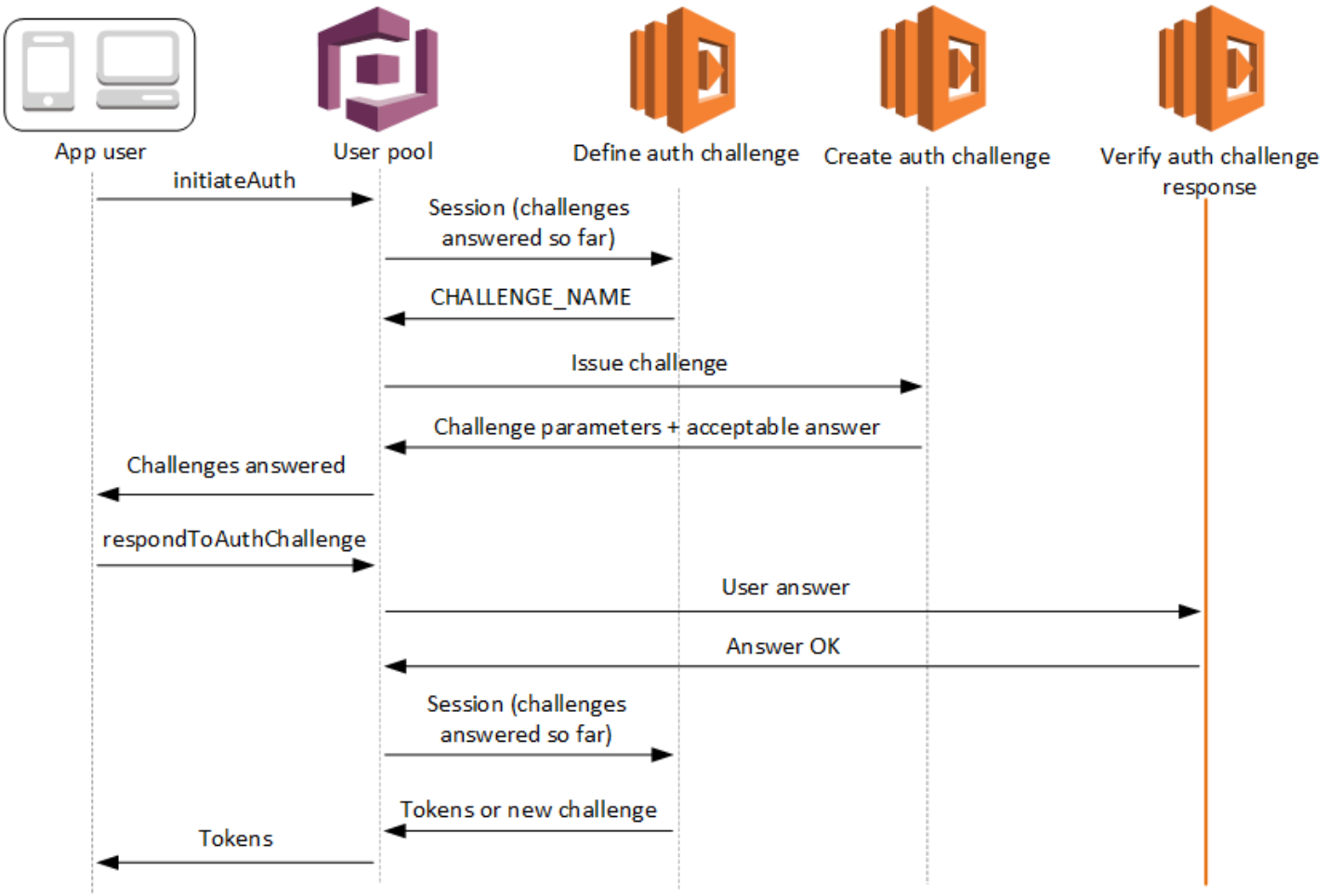
Un CAPTCHA viene creato come una sfida per l'utente. L'URL per l'immagine CAPTCHA viene aggiunto ai parametri di sfida pubblica come "`captchaUrl`", e la risposta prevista viene aggiunta ai parametri di sfida privata.

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 }
```

Trigger Lambda di una verifica di risposta di una richiesta di autenticazione



Verifica della risposta a una richiesta di autenticazione

Amazon Cognito richiama questo trigger per verificare se la risposta dell'utente finale per una richiesta di autenticazione personalizzata è valida o meno. Fa parte di un flusso di autenticazione personalizzato di un bacino d'utenza.

La richiesta per questo trigger include il `privateChallengeParameters` e la `challengeAnswer`. Il trigger Lambda di creazione di una richiesta di autenticazione restituisce valori `privateChallengeParameters` e include la risposta prevista fornita dall'utente. Il parametro `challengeAnswer` contiene la risposta dell'utente per la sfida.

La risposta contiene l'attributo `answerCorrect`. Se l'utente completa correttamente la sfida, Amazon Cognito imposta il valore dell'attributo su `true`. Se l'utente non completa correttamente la sfida, Amazon Cognito imposta il valore su `false`.

Il loop di sfide si ripete finché gli utenti non rispondono a tutte le sfide.

Argomenti

- Parametri del trigger Lambda della verifica di una richiesta di autenticazione
- Verifica di un esempio di risposta di una richiesta di autenticazione

Parametri del trigger Lambda della verifica di una richiesta di autenticazione

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        },
        "privateChallengeParameters": {
            "string": "string",
            . . .
        },
        "challengeAnswer": "string",
        "clientMetadata": {
```

```
            "string": "string",
            . . .
        },
        "userNotFound": boolean
    },
    "response": {
        "answerCorrect": boolean
    }
}
```

Parametri di richiesta di verifica sfida di autenticazione

userAttributes

Questo parametro contiene una o più coppie nome-valore che rappresentano gli attributi dell'utente.

userNotFound

Quando Amazon Cognito imposta `PreventUserExistenceErrors` su `ENABLED` per il tuo client del bacino d'utenza, Amazon Cognito popola questo valore booleano.

publicChallengeParameters

Questo parametro proviene dal trigger Creazione di una richiesta di autenticazione. Per determinare se l'utente ha superato una sfida, Amazon Cognito confronta i parametri con quelli della challengeAnswer di un utente.

Questo parametro contiene tutte le informazioni necessarie per convalidare la risposta dell'utente alla sfida. Tali informazioni includono la domanda che Amazon Cognito pone all'utente (`publicChallengeParameters`) e le risposte valide per la domanda (`privateChallengeParameters`). Questo parametro viene utilizzato solo dal trigger Lambda di verifica della risposta di una richiesta di autenticazione.

challengeAnswer

Il valore di questo parametro è la risposta dell'utente alla sfida.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di verifica di una richiesta di autenticazione. Per trasmettere questi dati alla funzione Lambda, utilizza il parametro ClientMetadata nelle operazioni API

AdminRespondToAuthChallenge e RespondToAuthChallenge. Quando Amazon Cognito trasmette la richiesta alla funzione di pre-generazione di token, non include dati del parametro ClientMetadata contenuti nelle operazioni API AdminInitiateAuth e InitiateAuth.

Parametri di risposta di verifica sfida di autenticazione

answerCorrect

Se l'utente completa correttamente la sfida, Amazon Cognito imposta questo parametro su `true`. Se l'utente non completa correttamente la sfida, Amazon Cognito imposta il parametro su `false`.

Verifica di un esempio di risposta di una richiesta di autenticazione

In questo esempio, la funzione Lambda verifica se la risposta dell'utente a una richiesta corrisponde alla risposta prevista. Se la risposta dell'utente corrisponde alla risposta prevista, il parametro `answerCorrect` è impostato su `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 };
```

Trigger Lambda di pre-generazione del token

Poiché Amazon Cognito richiama questo trigger prima della generazione di token, puoi personalizzare le richieste di token dei pool di utenti. Con le funzionalità di base dell'evento di attivazione della versione 1 o `V1_0` precedente alla generazione del token, puoi personalizzare il

token di identità (ID). Nei pool di utenti con funzionalità di sicurezza avanzate attive, puoi generare la versione 2 o `V2_0` dell'evento di attivazione con la personalizzazione del token di accesso.

Amazon Cognito invia un evento `V1_0` come richiesta alla funzione con dati che scriverebbe sul token ID. Un evento `V2_0` è una singola richiesta con i dati che Amazon Cognito scriverebbe sia nel token di identità che nel token di accesso. Per personalizzare entrambi i token, devi aggiornare la funzione in modo da utilizzare la versione trigger più recente e inviare i dati per entrambi i token nella stessa risposta.

Questo trigger Lambda può aggiungere, rimuovere e modificare alcune richieste nei token di identità e accesso prima che Amazon Cognito le invii all'app. Per usare questa funzionalità, puoi associare una funzione Lambda dalla console del pool di utenti di Amazon Cognito o aggiornare il pool di utenti `LambdaConfig` tramite la AWS Command Line Interface (AWS CLI).

Versioni degli eventi

Il tuo pool di utenti può fornire diverse versioni di un evento di attivazione precedente alla generazione di token alla tua funzione Lambda. Un `V1_0` trigger fornisce i parametri per la modifica dei token ID. Un `V2_0` trigger fornisce i parametri per quanto segue.

1. Le funzioni di un `V1_0` trigger.
2. La possibilità di personalizzare i token di accesso.
3. La possibilità di passare tipi di dati complessi a ID e accedere ai valori delle richieste di token:
   - Stringa
   - Numero
   - Booleano
   - Matrice di stringhe, numeri, valori booleani o una combinazione di questi
   - JSON

> **Note**
>
> Nel token ID, è possibile compilare oggetti complessi in base ai valori delle attestazioni ad eccezione di`phone_number_verified`,, `email_verified` e. `updated_at` `address`

I pool di utenti forniscono `V1_0` eventi per impostazione predefinita. Per configurare il tuo pool di utenti per l'invio di un `V2_0` evento, scegli una versione Trigger event delle funzionalità di Basic +

personalizzazione del token di accesso quando configuri il trigger nella console Amazon Cognito. Puoi anche impostare il valore di `LambdaVersion` nei LambdaConfigparametri in una richiesta UpdateUserPool o CreateUserPool API. La personalizzazione dei token di accesso con `V2_0` eventi comporta costi aggiuntivi. Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Attestazioni e ambiti esclusi

Amazon Cognito limita le attestazioni e gli ambiti che puoi aggiungere, modificare o sopprimere nei token di accesso e identità. Se la funzione Lambda tenta di impostare un valore per una di queste attestazioni, Amazon Cognito emette un token con il valore dell'attestazione originale, se presente nella richiesta.

Attestazioni condivise

- `acr`
- `amr`
- `at_hash`
- `auth_time`
- `azp`
- `exp`
- `iat`
- `iss`
- `jti`
- `nbf`
- `nonce`
- `origin_jti`
- `sub`
- `token_use`

Attestazioni relative ai token ID

- `identities`
- `aud`
- `cognito:username`

Attestazioni relative ai token di accesso

- `username`
- `client_id`
- `scope`

> **Note**
>
> Puoi modificare gli ambiti in un token di accesso con valori di risposta `scopesToAdd` e `scopesToSuppress`, ma non puoi modificare direttamente l'attestazione `scope`. Non è possibile aggiungere ambiti che iniziano con `aws.cognito`, incluso l'ambito riservato dei pool di utenti `aws.cognito.signin.user.admin`.

- `device_key`
- `event_id`
- `version`

Non è possibile aggiungere o sostituire le attestazioni con i seguenti prefissi, ma è possibile sopprimerle o impedire che vengano visualizzate nel token.

- `dev:`
- `cognito:`

Puoi aggiungere un'attestazione `aud` ai token di accesso, ma il relativo valore deve corrispondere all'ID client dell'app della sessione corrente. Puoi derivare l'ID del client nell'evento di richiesta da `event.callerContext.clientId`.

Personalizzazione del token di identità

Con il trigger Lambda di pre-generazione di token, puoi personalizzare il contenuto di un token ID dal pool di utenti. Il token ID fornisce gli attributi utente da un'origine di identità affidabile per l'accesso a un'app web o per dispositivi mobili. Per ulteriori informazioni sui token ID, consulta Utilizzo di token ID.

Gli usi del trigger Lambda di pre-generazione di token con un token ID includono quanto segue.

- Apportare una modifica in fase di runtime al ruolo IAM richiesto dall'utente da un pool di identità.

- Aggiungere gli attributi utente da un'origine esterna.
- Aggiungere o sostituire i valori degli attributi utente esistenti.
- Sopprimere la divulgazione degli attributi utente che, a causa degli ambiti autorizzati dell'utente e dell'accesso in lettura agli attributi concesso al client dell'app, verrebbero altrimenti trasferiti all'app.

Personalizzazione del token di accesso

Con il trigger Lambda di pre-generazione di token, puoi personalizzare il contenuto di un token di accesso dal pool di utenti. Il token di accesso autorizza gli utenti a recuperare informazioni da risorse ad accesso protetto come operazioni API autorizzate da token Amazon Cognito e API di terze parti. Sebbene sia possibile generare token di accesso per l'autorizzazione machine-to-machine (M2M) con Amazon Cognito con una concessione di credenziali client, le richieste M2M non richiamano la funzione di attivazione precedente alla generazione di token e non possono emettere token di accesso personalizzati. Per ulteriori informazioni sui token di accesso, consulta Utilizzo del token di accesso.

Gli usi del trigger Lambda di pre-generazione di token con un token di accesso includono i seguenti.

- Aggiungere o sopprimere gli ambiti OAuth 2.0 nell'attestazione `scope`. Ad esempio, puoi aggiungere ambiti a un token di accesso ottenuto dall'autenticazione API dei pool di utenti di Amazon Cognito, che assegna solo l'ambito `aws.cognito.signin.user.admin`.
- Modificare l'appartenenza di un utente ai gruppi di pool di utenti.
- Aggiunta di richieste non ancora presenti in un token di accesso Amazon Cognito.
- Bloccare la divulgazione di attestazioni che altrimenti verrebbero trasmesse all'app.

Per supportare la personalizzazione dell'accesso nel pool di utenti, è necessario configurare il pool di utenti per generare una versione aggiornata della richiesta di trigger. Aggiornare il pool di utenti come illustrato nella procedura seguente.

AWS Management Console

Per supportare la personalizzazione del token di accesso in un trigger Lambda di pre-generazione di token

1. Passa alla console Amazon Cognito e scegli User Pools (Bacini d'utenza).
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.

3. Se non l'hai già fatto, attiva le funzionalità di sicurezza avanzate dalla scheda Integrazione app.
4. Scegli la scheda Proprietà del bacino d'utenza e individua Lambda triggers (Trigger Lambda).
5. Aggiungi o modifica un Trigger di pre-generazione di token.
6. Scegli una funzione Lambda in Assegna la funzione Lambda.
7. Scegli una Versione dell'evento trigger delle funzionalità di base + personalizzazione del token di accesso. Questa impostazione aggiorna i parametri di richiesta inviati da Amazon Cognito alla funzione per includere campi per la personalizzazione del token di accesso.

User pools API

Per supportare la personalizzazione del token di accesso in un trigger Lambda di pre-generazione di token

UpdateUserPoolGenera una richiesta o API. CreateUserPool È necessario specificare un valore per tutti i parametri che non si desidera impostare su un valore predefinito. Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti.

Includi il seguente contenuto nel parametro `LambdaVersion` della richiesta. Un valore `LambdaVersion` di `V2_0` fa sì che il pool di utenti aggiunga parametri per la personalizzazione del token di accesso. Per richiamare una versione di funzione specifica, usa un ARN della funzione Lambda on una versione della funzione come valore di `LambdaArn`.

```
"PreTokenGenerationConfig": {
   "LambdaArn": "arn:aws:lambda:us-west-2:123456789012:function:MyFunction",
   "LambdaVersion": "V2_0"
},
```

Argomenti

- Origini del trigger Lambda di pre-generazione del token
- Parametri del trigger Lambda di pre-generazione del token
- Esempio della versione due dell'evento trigger pre-token: aggiunta ed eliminazione di richieste, ambiti e gruppi
- Evento precedente alla generazione di token (versione due): aggiungere attestazioni con oggetti complessi

- Esempio di evento versione 1 di generazione del pre-token: aggiunta di una nuova richiesta ed eliminazione di una richiesta esistente
- Esempio di evento versione 1 di generazione del pre-token: modifica dell'appartenenza al gruppo dell'utente

Origini del trigger Lambda di pre-generazione del token

Valore triggerSource	Evento
`TokenGeneration_HostedAuth`	Richiamato durante l'autenticazione dalla pagina di accesso dell'interfaccia utente ospitata di Amazon Cognito.
`TokenGeneration_Authentication`	Chiamato dopo che i flussi di autenticazione degli utenti sono terminati.
`TokenGeneration_NewPasswordChallenge`	Chiamato dopo che l'utente viene creato da un amministratore. Questo flusso viene chiamato quando l'utente deve cambiare una password temporanea.
`TokenGeneration_AuthenticateDevice`	Chiamato alla fine dell'autenticazione di un dispositivo dell'utente.
`TokenGeneration_RefreshTokens`	Chiamato quando un utente cerca di aggiornare l'identità e i token di accesso.

Parametri del trigger Lambda di pre-generazione del token

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste. Quando aggiungi un trigger Lambda di pre-generazione di token al pool di utenti, puoi scegliere una versione del trigger. Questa versione determina se Amazon Cognito passa una richiesta alla funzione Lambda con parametri aggiuntivi per la personalizzazione del token di accesso.

Version 1

Il token versione 1 può impostare l'appartenenza al gruppo, i ruoli IAM e nuove attestazioni nei token ID.

```
{
    "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'evento di richiesta della versione 2 aggiunge campi che personalizzano il token di accesso. Aggiunge inoltre il supporto per tipi di `claimsToOverride` dati complessi nell'oggetto di risposta. La tua funzione Lambda può restituire i seguenti tipi di dati nel valore di: `claimsToOverride`

- Stringa
- Numero
- Booleano
- Matrice di stringhe, numeri, valori booleani o una combinazione di questi
- 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"
            }
        }
    }
}
```

Parametri di richiesta di pre-generazione del token

Nome	Descrizione	Versione minima dell'evento di trigger
userAttributes	Attributi del profilo utente nel pool di utenti.	1
groupConfiguration	L'oggetto di input contenente l'attuale configurazione del gruppo. L'oggetto include `groupsToOverride` , `iamRolesToOverride` `epreferredRole` .	1
groupsToOverride	I gruppi di pool di utenti di cui l'utente fa parte.	1
iamRolesToSostituisci	È possibile associare un gruppo di pool di utenti a un ruolo AWS Identity and Access Management (IAM). Questo elemento è un elenco di tutti i ruoli IAM dei gruppi di cui l'utente è membro.	1
preferredRole	Puoi impostare una precedenza per i gruppi di pool di utenti. Questo elemento contiene il nome del ruolo IAM del gruppo con la precedenza più alta nell'elemento `groupsToOverride` .	1

Nome	Descrizione	Versione minima dell'evento di trigger
clientMetadata	Una o più coppie chiave-valore che è possibile specificare e fornire come input personalizzato alla funzione Lambda per il trigger di pre-generazione del token. Per passare questi dati alla funzione Lambda, usa il ClientMetadata parametro nelle operazioni AdminRespondToAuthChallenge e RespondToAuthChallengeAPI. Amazon Cognito non include i dati del `ClientMetadata` parametro AdminInitiateAuth e le operazioni InitiateAuthAPI nella richiesta che passa alla funzione di generazione precedente al token.	1
ambiti	Ambiti OAuth 2.0 dell'utente. Gli ambiti presenti in un token di accesso sono gli ambiti standard e personalizzati del pool di utenti richiesti dall'utente e che il client dell'app è autorizzato a emettere.	2

Parametri di risposta di pre-generazione del token

Nome	Descrizione	Versione minima dell'evento di trigger
claimsOverrideDetails	Un container per tutti gli elementi di un evento trigger `V1_0`.	1
claimsAndScopeOverrideDetails	Un container per tutti gli elementi di un evento trigger `V2_0`.	2
idTokenGeneration	Le attestazioni che desideri sostituire, aggiungere o sopprimere nel token di identità dell'utente. Questo elemento padre ai valori di personalizzazione del token ID	2

Nome	Descrizione	Versione minima dell'evento di trigger
	viene visualizzato solo negli eventi della versione 2, ma gli elementi secondari vengono visualizzati negli eventi della versione 1.	
accessTokenGeneration	Le attestazioni e gli ambiti che desideri sostituire, aggiungere o sopprimere nel token di accesso dell'utente. Questo elemento padre per accedere ai valori di personalizzazione del token viene visualizzato solo negli eventi della versione 2.	2
claimsToAddOrOverride	Una mappa di una o più attestazioni e dei relativi valori che desideri aggiungere o modificare. Per attestazioni relative ai gruppi, utilizza invece `groupOverrideDetails` . Negli eventi della versione 2, questo elemento viene visualizzato sotto `accessTokenGeneration` e `idTokenGeneration` .	1 *
claimsToSuppress	Un elenco delle attestazioni che desideri vengano soppresse da Amazon Cognito. Se la tua funzione sopprime e sostituisce un valore di attestazione, Amazon Cognito sopprime l'attestazione. Negli eventi della versione 2, questo elemento viene visualizzato sotto `accessTokenGeneration` e `idTokenGeneration` .	1

Nome	Descrizione	Versione minima dell'evento di trigger
groupOverrideDetails	L'oggetto di output contenente l'attuale configurazione di gruppo. L'oggetto include `groupsToOverride` , `iamRolesToOverride` e `preferredRole` . La funzione sostituisce l'oggetto `groupOverrideDetails` con l'oggetto fornito. Se fornisci un oggetto vuoto o nullo nella risposta, Amazon Cognito sopprime i gruppi. Per mantenere la configurazione di un gruppo esistente così com'è, copia il valore dell'oggetto `groupConfiguration` della richiesta nell'oggetto `groupOverrideDetails` della risposta. Quindi ritrasmettilo al servizio. Amazon Cognito ID e i token di accesso contengono entrambi i token reclamo `cognito:groups` . L'oggetto `groupOverrideDetails` sostituirà l'attestazione `cognito:groups` nei token di accesso e nei token ID.	1
scopesToAdd	Un elenco di ambiti OAuth 2.0 che desideri aggiungere all'attestazione `scope` nel token di accesso dell'utente. Non puoi aggiungere valori di ambito che contengono uno o più caratteri con spazi vuoti.	2
scopesToSuppress	Un elenco di ambiti OAuth 2.0 che desideri rimuovere dall'attestazione `scope` nel token di accesso dell'utente.	2

* Gli oggetti di risposta agli eventi della versione 1 possono restituire stringhe. Gli oggetti di risposta agli eventi della versione 2 possono restituire oggetti complessi.

Esempio della versione due dell'evento trigger pre-token: aggiunta ed eliminazione di richieste, ambiti e gruppi

Questo esempio apporta le seguenti modifiche ai token di un utente.

1. Imposta i `family_name` valori come `Doe` nel token ID.
2. Impedisce la visualizzazione delle richieste `email` e `phone_number` nel token ID.
3. Imposta la richiesta `cognito:roles` del token ID su `"arn:aws:iam::123456789012:role\/sns_callerA","arn:aws:iam::123456789012:role\/sns_callerC","arn:aws:iam::123456789012:role\/sns_callerB"`.
4. Imposta la richiesta `cognito:preferred_role` del token ID su `arn:aws:iam::123456789012:role/sns_caller`.
5. Aggiunge gli ambiti `openid`, `email` e `solar-system-data/asteroids.add` al token di accesso.
6. Elimina l'ambito `phone_number` e `aws.cognito.signin.user.admin` dal token di accesso. La rimozione di `phone_number` impedisce il recupero del numero di telefono dell'utente da `userInfo`. La rimozione di `aws.cognito.signin.user.admin` impedisce alle richieste API da parte dell'utente di leggere e modificare il proprio profilo con l'API dei pool di utenti di Amazon Cognito.

   > **Note**
   >
   > La rimozione di `phone_number` dagli ambiti impedisce solo il recupero del numero di telefono di un utente se gli ambiti rimanenti nel token di accesso includono `openid` e almeno un altro ambito standard. Per ulteriori informazioni, consulta Informazioni sugli ambiti.

7. Imposta la richiesta `cognito:groups` del token ID e di accesso su `"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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

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": []
    }
}
```

Evento precedente alla generazione di token (versione due): aggiungere attestazioni con oggetti complessi

Questo esempio apporta le seguenti modifiche ai token di un utente.

1. Aggiunge rivendicazioni di tipo number, string, booleano e JSON al token ID. Questa è l'unica modifica che gli eventi di attivazione della versione due rendono disponibile al token ID.
2. Aggiunge rivendicazioni di tipo number, string, boolean e JSON al token di accesso.

3. Aggiunge tre ambiti al token di accesso.
4. Elimina le `sub` affermazioni `email` e nell'ID e nei token di accesso.
5. Sopprime l'`aws.cognito.signin.user.admin`ambito nel token di accesso.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

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": []
    }
}
```

Esempio di evento versione 1 di generazione del pre-token: aggiunta di una nuova richiesta ed eliminazione di una richiesta esistente

Questo esempio usa un evento trigger versione 1 con una funzione Lambda di pre-generazione di token per aggiungere una nuova attestazione ed eliminarne una esistente.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per il codice di esempio. Dal momento che il codice di esempio non elabora i parametri di richiesta, puoi utilizzare un evento di test con una richiesta vuota. Per ulteriori informazioni sui parametri di richiesta comuni, consulta Evento trigger Lambda per il bacino d'utenza.

JSON

```
{
  "request": {},
  "response": {}
}
```

Esempio di evento versione 1 di generazione del pre-token: modifica dell'appartenenza al gruppo dell'utente

Questo esempio usa un evento trigger versione 1 con una funzione Lambda di pre-generazione di token per modificare l'appartenenza ai gruppi dell'utente.

Node.js

```
const handler = async (event) => {
  event.response = {
    claimsOverrideDetails: {
      groupOverrideDetails: {
        groupsToOverride: ["group-A", "group-B", "group-C"],
```

```
        iamRolesToOverride: [
          "arn:aws:iam::XXXXXXXXXXX:role/sns_callerA",
          "arn:aws:iam::XXXXXXXXX:role/sns_callerB",
          "arn:aws:iam::XXXXXXXXXX:role/sns_callerC",
        ],
        preferredRole: "arn:aws:iam::XXXXXXXXXXX:role/sns_caller",
      },
    },
  };

  return event;
};

export { handler };
```

Amazon Cognito trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

JSON

```
{
  "request": {},
  "response": {}
}
```

Trigger Lambda di migrazione utenti

Amazon Cognito richiama questo trigger quando un utente non esiste nel bacino d'utenza al momento dell'accesso con una password oppure nel flusso di reimpostazione della password. Dopo l'esito positivo della funzione Lambda, Amazon Cognito crea l'utente nel bacino d'utenza. Per i dettagli sul flusso di autenticazione con il trigger Lambda di migrazione utenti, consulta Importazione di utenti in bacini d'utenza con un trigger Lambda di migrazione utenti.

Usa questo trigger Lambda per migrare gli utenti dalla directory degli utenti esistente nel bacino d'utenza ad Amazon Cognito al momento dell'accesso o durante il flusso di reimpostazione della password.

Argomenti

- Origini dei trigger Lambda di migrazione utenti
- Parametri del trigger Lambda di migrazione utenti
- Esempio di migrazione di un utente con una password esistente

Origini dei trigger Lambda di migrazione utenti

Valore triggerSource	Evento
`UserMigration_Authentication`	Migrazione degli utenti al momento dell'acce sso.
`UserMigration_ForgotPassword`	Migrazione degli utenti durante il flusso di password dimenticata.

Parametri del trigger Lambda di migrazione utenti

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

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
    }
}
```

Parametri di richiesta di migrazione utenti

userName

Il nome utente inserito dall'utente al momento dell'accesso.

password

La password inserita dall'utente al momento dell'accesso. Amazon Cognito non invia questo valore in una richiesta avviata da un flusso di reimpostazione della password.

validationData

Una o più coppie chiave-valore contenente i dati di convalida nella richiesta di registrazione dell'utente. Per trasmettere questi dati alla funzione Lambda, usa il parametro ClientMetadata nelle operazioni API InitiateAuth e AdminInitiateAuth.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda per il trigger di migrazione utenti. Puoi trasmettere questi dati alla funzione Lambda utilizzando il parametro ClientMetadata nelle operazioni API AdminRespondToAuthChallenge e RespondToAuthChallenge.

Parametri di risposta di migrazione utenti

userAttributes

Questo campo è obbligatorio.

Questo campo deve contenere una o più coppie nome-valore archiviate da Amazon Cognito nel profilo utente del bacino d'utenza e utilizzate come attributi utente. Puoi includere attributi utenti

standard e personalizzati. Gli attributi personalizzati richiedono il prefisso `custom:` per distinguerli dagli attributi standard. Per ulteriori informazioni sugli attributi, consulta Attributi personalizzati.

> **Note**
>
> Per poter reimpostare le password nel flusso di reimpostazione della password, gli utenti devono avere un indirizzo e-mail o un numero di telefono verificati. Amazon Cognito invia un messaggio contenente un codice di reimpostazione della password all'indirizzo e-mail o al numero di telefono specificati negli attributi utente.

Attributi	Requisito
Tutti gli attributi contrassegnati come obbligatori al momento della creazione del bacino d'utenza	Se durante la migrazione manca uno qualsiasi degli attributi obbligatori, Amazon Cognito utilizza valori di default.
`username`	Obbligatorio se il bacino d'utenza è stato configurato con attributi alias per l'accesso, oltre al nome utente, e l'utente ha inserito come nome utente un valore alias valido. Questo valore alias può essere un indirizzo e-mail, un nome utente preferito o un numero di telefono. Se la richiesta e il bacino d'utenza soddisfano i requisiti di alias, la risposta della funzione deve assegnare il parametro `username` che ha ricevuto a un attributo alias. Inoltre, la risposta deve assegnare il tuo valore all'attributo `username`. Se il bacino d'utenza non soddisfa le condizioni richieste per mappare lo `username` ricevuto a un alias, il parametro `username` nella risposta deve corrispon dere esattamente alla richiesta o essere omesso. **Note** Lo `username` deve essere univoco all'interno del bacino d'utenti.

finalUserStatus

È possibile impostare questo parametro su `CONFIRMED` per confermare automaticamente gli utenti, in modo che possano accedere con le password precedenti. Gli utenti che imposti su `CONFIRMED` non devono intraprendere ulteriori azioni prima di poter effettuare l'accesso. Se non imposti questo attributo su `CONFIRMED`, esso rimane impostato su `RESET_REQUIRED`.

Un `finalUserStatus` di `RESET_REQUIRED` significa che l'utente deve modificare la propria password immediatamente dopo la migrazione al momento dell'accesso e l'app client deve gestire l'eccezione `PasswordResetRequiredException` durante il flusso di autenticazione.

> **Note**
>
> Amazon Cognito non applica la policy relativa alla forza della password che hai configurato per il bacino d'utenza durante la migrazione utilizzando il trigger Lambda. Se la password non soddisfa la tua policy per le password, Amazon Cognito la accetta comunque e continua a migrare l'utente. Per applicare la policy di protezione delle password e rifiutare le password che non soddisfano la policy, convalida la validità della password nel codice. Quindi, se la password non soddisfa la policy, imposta finalUserStatus su `RESET_REQUIRED`.

messageAction

Se desideri impedire l'invio del messaggio di benvenuto che Amazon Cognito invia di solito ai nuovi utenti, imposta questo parametro su `SUPPRESS`. Se la funzione non restituisce questo parametro, Amazon Cognito invia il messaggio di benvenuto.

desiredDeliveryMediums

Per inviare il messaggio di benvenuto tramite e-mail, imposta il valore su `EMAIL`; per inviarlo tramite SMS, imposta il valore su `SMS`. Se la funzione non restituisce questo parametro, Amazon Cognito invia il messaggio di benvenuto tramite SMS.

forceAliasCreation

Se imposti questo parametro su `TRUE` e il numero di telefono o l'indirizzo e-mail specificato nel parametro UserAttributes esiste già come alias per un altro utente, la chiamata API migra l'alias dall'utente precedente all'utente appena creato. L'utente precedente non potrà più effettuare l'accesso utilizzando tale alias.

Se imposti questo parametro su `FALSE` e l'alias esiste, Amazon Cognito non esegue la migrazione dell'utente e restituisce un errore all'app client.

Se non restituisci questo parametro, Amazon Cognito presuppone che il suo valore sia "false".

enableSMSMFA

Imposta questo parametro su `true` per richiedere all'utente migrato di completare l'autenticazione a più fattori (MFA) tramite SMS per accedere. Il pool di utenti deve avere l'MFA abilitato. Gli attributi dell'utente nei parametri della richiesta devono includere un numero di telefono, altrimenti la migrazione di quell'utente fallirà.

Esempio di migrazione di un utente con una password esistente

Questa funzione Lambda di esempio migra l'utente con una password esistente ed elimina il messaggio di benvenuto da 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 };
```

Trigger Lambda di messaggi personalizzati

Amazon Cognito richiama questo trigger prima di inviare un messaggio di verifica via e-mail o telefono o un codice di autenticazione a più fattori (MFA). È possibile personalizzare il messaggio in modo dinamico con il trigger di messaggio personalizzato. Per modificare i messaggi personalizzati statici, vai alla scheda Message customizations (Personalizzazioni del messaggio) della console originale di Amazon Cognito.

La richiesta include `codeParameter`. Questa è una stringa che agisce da segnaposto per il codice che Amazon Cognito fornisce all'utente. Inserisci la stringa `codeParameter` nel corpo del messaggio, nella posizione in cui desideri che appaia il codice di verifica. Alla ricezione di questa risposta, Amazon Cognito sostituisce la stringa `codeParameter` con il codice di verifica effettivo.

> **Note**
>
> Una funzione Lambda di messaggio personalizzato con l'origine di trigger `CustomMessage_AdminCreateUser` restituisce un nome utente e un codice di verifica. Poiché un utente creato dall'amministratore deve ricevere sia il proprio nome utente che il codice, la risposta della funzione deve includere sia `request.usernameParameter` che `request.codeParameter`.

Argomenti

- Origini dei trigger Lambda di messaggi personalizzati
- Parametri del trigger Lambda di messaggi personalizzati
- Messaggio personalizzato per un esempio di registrazione
- Messaggio personalizzato per un esempio di amministratore che crea un utente

Origini dei trigger Lambda di messaggi personalizzati

Valore triggerSource	Evento
`CustomMessage_SignUp`	Messaggio personalizzato: invio del codice di conferma post registrazione.
`CustomMessage_AdminCreateUser`	Messaggio personalizzato: invio della password temporanea a un nuovo utente.
`CustomMessage_ResendCode`	Messaggio personalizzato: nuovo invio del codice di conferma a un utente esistente.
`CustomMessage_ForgotPassword`	Messaggio personalizzato: invio del codice di conferma per una richiesta di password dimenticata.
`CustomMessage_UpdateUserAttribute`	Messaggio personalizzato: quando l'e-mail o il numero di telefono di un utente viene modificato, questo trigger invia automaticamente un

Valore triggerSource	Evento
	codice di verifica all'utente. Non può essere utilizzato per altri attributi.
`CustomMessage_VerifyUserAttribute`	Messaggio personalizzato: questo trigger invia un codice di verifica all'utente, quando lo richiede manualmente per una nuova e-mail o numero di telefono.
`CustomMessage_Authentication`	Messaggio personalizzato: invio del codice MFA durante l'autenticazione.

Parametri del trigger Lambda di messaggi personalizzati

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "userAttributes": {
            "string": "string",
            . . .
        }
        "codeParameter": "####",
        "usernameParameter": "string",
        "clientMetadata": {
            "string": "string",
            . . .
        }
    },
    "response": {
        "smsMessage": "string",
        "emailMessage": "string",
        "emailSubject": "string"
    }
}
```

Parametri di richiesta di messaggi personalizzati

userAttributes

Una o più coppie nome-valore che rappresentano gli attributi utente.

codeParameter

Una stringa per te, da utilizzare come segnaposto per il codice di verifica nel messaggio personalizzato.

Parametro username

Il nome utente. Amazon Cognito include questo parametro nelle richieste generate da utenti creati dall'amministratore.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato alla funzione Lambda specificata per il trigger di messaggi personalizzato. La richiesta che richiama una funzione di messaggio personalizzata non include i dati passati nel ClientMetadata parametro AdminInitiateAuthe le operazioni InitiateAuthAPI. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle seguenti azioni API:

- AdminResetUserPassword
- AdminRespondToAuthChallenge
- AdminUpdateUserAttributes
- ForgotPassword
- GetUserAttributeVerificationCode
- ResendConfirmationCode
- SignUp
- UpdateUserAttributes

Parametri di risposta di messaggi personalizzati

Nella risposta, specifica il testo personalizzato da usare nei messaggi destinati agli utenti. Per i vincoli di stringa che Amazon Cognito applica a questi parametri, consulta. MessageTemplateType

smsMessage

L'SMS personalizzato da inviare ai tuoi utenti. Deve includere il valore `codeParameter` che hai ricevuto nella richiesta.

emailMessage

Il messaggio e-mail personalizzato da inviare ai tuoi utenti. È possibile utilizzare la formattazione HTML nel parametro `emailMessage`. Deve includere il valore `codeParameter` ricevuto nella richiesta come variabile `{####}`. Amazon Cognito può utilizzare il parametro `emailMessage` solo se l'attributo `EmailSendingAccount` del bacino d'utenza è `DEVELOPER`. Se l'attributo `EmailSendingAccount` del bacino d'utenza non è `DEVELOPER` e viene restituito un parametro `emailMessage`, Amazon Cognito genera un codice di errore 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. Quando scegli di utilizzare Amazon Simple Email Service (Amazon SES) per inviare messaggi e-mail, l'attributo `EmailSendingAccount` di un bacino d'utenza è `DEVELOPER`. In caso contrario, il valore è `COGNITO_DEFAULT`.

emailSubject

L'oggetto per il messaggio personalizzato. Puoi utilizzare il `emailSubject` parametro solo se l' EmailSendingAccountattributo del pool di utenti è. `DEVELOPER` Se l'attributo `EmailSendingAccount` del bacino d'utenza non è `DEVELOPER` e viene restituito un parametro `emailSubject`, Amazon Cognito genera un codice di errore 400 `com.amazonaws.cognito.identity.idp.model.InvalidLambdaResponseException`. Quando scegli di utilizzare Amazon Simple Email Service (Amazon SES) per inviare messaggi e-mail, l'attributo `EmailSendingAccount` di un bacino d'utenza è `DEVELOPER`. In caso contrario, il valore è `COGNITO_DEFAULT`

Messaggio personalizzato per un esempio di registrazione

Questa funzione Lambda di esempio permette di personalizzare un'e-mail o un SMS quando il servizio richiede che un'app invii un codice di verifica all'utente.

Amazon Cognito può richiamare un trigger Lambda durante più eventi: dopo la registrazione, per rinviare un codice di verifica, nel caso di una password dimenticata o per verificare un attributo utente. La risposta include sia SMS sia e-mail. Il messaggio deve includere il parametro di codice `"####"`. Questo parametro è il segnaposto per il codice di verifica che l'utente riceve.

La lunghezza massima per un messaggio e-mail è di 20.000 caratteri UTF-8. Questa lunghezza include il codice di verifica. In questi messaggi e-mail puoi utilizzare i tag HTML.

La lunghezza massima di un SMS è di 140 caratteri UTF-8. Questa lunghezza include il codice di verifica.

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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

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>"
    }
}
```

Messaggio personalizzato per un esempio di amministratore che crea un utente

La richiesta che Amazon Cognito ha inviato a questo esempio di messaggio personalizzato (funzione Lambda) ha un `triggerSource` valore di, un nome utente `CustomMessage_AdminCreateUser` e una password temporanea. La funzione viene compilata `${event.request.codeParameter}` a partire dalla password temporanea nella richiesta e `${event.request.usernameParameter}` dal nome utente nella richiesta.

I messaggi personalizzati devono inserire i valori di `codeParameter` e `usernameParameter` dentro `smsMessage` e `emailMessage` nell'oggetto di risposta. In questo esempio, la funzione scrive lo stesso messaggio nei campi di risposta `event.response.smsMessage` e`event.response.emailMessage`.

La lunghezza massima di un messaggio e-mail è pari a 20.000 caratteri UTF-8. Questa lunghezza include il codice di verifica. Puoi usare i tag HTML in queste e-mail. La lunghezza massima di un SMS è di 140 caratteri UTF-8. Questa lunghezza include il codice di verifica.

La risposta include sia SMS sia 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 trasferisce informazioni sugli eventi alla funzione Lambda. La funzione quindi restituisce a Amazon Cognito lo stesso oggetto evento con eventuali modifiche nella risposta. Nella console Lambda puoi configurare un evento di test con i dati pertinenti al trigger Lambda. Di seguito è riportato un evento di test per questo esempio di codice:

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>"
```

```
  }
}
```

Trigger Lambda del mittente personalizzato

I bacini d'utenza di Amazon Cognito forniscono i trigger Lambda `CustomEmailSender` e `CustomSMSSender` per attivare notifiche e-mail e SMS di terze parti. Per inviare notifiche agli utenti dall'interno del codice della funzione Lambda, puoi utilizzare i provider di SMS ed e-mail che desideri. Quando Amazon Cognito deve inviare agli utenti notifiche come codici di conferma, codici di verifica o password temporanee, gli eventi attivano le funzioni Lambda configurate. Amazon Cognito invia il codice e le password temporanee (segreti) alle funzioni Lambda attivate. Amazon Cognito crittografa questi segreti con una chiave AWS KMS gestita dal cliente e il AWS Encryption SDK. AWS Encryption SDK è una libreria di crittografia lato client che ti consente di crittografare e decrittografare i dati generici.

> **Note**
>
> Per configurare i bacini d'utenza affinché utilizzino questi trigger Lambda, puoi utilizzare la AWS CLI o l'SDK. Queste configurazioni non sono disponibili nella console Amazon Cognito.

CustomEmailSender

Amazon Cognito richiama questo trigger per inviare notifiche e-mail agli utenti.

CustomSMSSender

Amazon Cognito richiama questo trigger per inviare notifiche SMS agli utenti.

Risorse

Le seguenti risorse possono aiutarti a utilizzare i trigger `CustomEmailSender` e `CustomSMSSender`.

AWS KMS

AWS KMS è un servizio gestito per creare e controllare chiavi AWS KMS. Queste chiavi crittografano i tuoi dati. Per ulteriori informazioni, consulta la pagina Che cos'è AWS Key Management Service?

Chiave KMS

Una chiave KMS è la rappresentazione logica di una chiave crittografica. La chiave KMS include metadati, ad esempio l'ID della chiave, la data di creazione, la descrizione e lo stato della chiave. La chiave KMS contiene anche il materiale della chiave utilizzato per crittografare e decrittare i dati. Per ulteriori informazioni, consulta AWS Chiavi KMS.

Chiavi KMS simmetriche

Una chiave KMS simmetrica è una chiave di crittografia a 256 bit che non mantiene crittografato AWS KMS. Per utilizzare una chiave KMS simmetrica è necessario chiamare AWS KMS. Amazon Cognito utilizza le chiavi simmetriche. La stessa chiave viene usata per la crittografia e la decrittografia. Per ulteriori informazioni, consulta Chiavi KMS simmetriche.

Trigger Lambda del mittente di e-mail personalizzato

Quando assegni un trigger del mittente dell'e-mail personalizzato al pool di utenti, Amazon Cognito richiama una funzione Lambda anziché il suo comportamento predefinito quando un evento utente richiede l'invio di un messaggio e-mail. Con un trigger del mittente personalizzato, la funzione AWS Lambda può inviare notifiche e-mail agli utenti tramite un metodo e un provider di propria scelta. Il codice personalizzato della funzione deve elaborare e distribuire tutti i messaggi e-mail del pool di utenti.

> **Note**
>
> Al momento, non è possibile assegnare trigger del mittente personalizzati nella console Amazon Cognito. Puoi assegnare un trigger con il parametro `LambdaConfig` in una richiesta API `CreateUserPool` o `UpdateUserPool`.

Per usare questo trigger, esegui questi passaggi:

1. Crea una chiave di crittografia simmetrica in AWS Key Management Service (AWS KMS). Amazon Cognito genera segreti (password temporanee, codici di verifica e codici di conferma), quindi utilizza questa chiave KMS per crittografarli. Puoi quindi usare l'operazione API Decrittografa nella funzione Lambda per decrittografare i segreti e inviarli all'utente come testo non crittografato. Il AWS Encryption SDK è uno strumento utile per operazioni AWS KMS nella tua funzione.
2. Crea una funzione Lambda che desideri assegnare come trigger del mittente personalizzato. Concedi al ruolo della funzione Lambda autorizzazioni `kms:Decrypt` per la chiave KMS.

3. Concedi l'accesso `cognito-idp.amazonaws.com` al principale del servizio Amazon Cognito per richiamare la funzione Lambda.
4. Scrivi il codice della funzione Lambda che indirizza i messaggi ai metodi di distribuzione personalizzati o a fornitori di terze parti. Per distribuire il codice di verifica o conferma dell'utente, Base64 decodifica e decrittografa il valore del parametro `code` nella richiesta. Questa operazione produce un codice o una password in testo normale da includere nel messaggio.
5. Aggiorna il bacino d'utenza in modo che utilizzi un trigger Lambda del mittente personalizzato. Il principale IAM che aggiorna o crea un pool di utenti con un trigger del mittente personalizzato deve disporre dell'autorizzazione per creare una concessione per la tua chiave KMS. Il seguente frammento `LambdaConfig` assegna funzioni del mittente SMS ed e-mail personalizzate.

```
"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"
   }
```

Parametri del trigger Lambda del mittente di e-mail personalizzato

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "type": "customEmailSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
```

```
            . . .
        }
}
```

Parametri della richiesta del mittente di e-mail personalizzato

type

La versione della richiesta. Per un evento del mittente di e-mail personalizzato, il valore di questa stringa è sempre `customEmailSenderRequestV1`.

code

Il codice crittografato che la funzione può decrittografare e inviare all'utente.

clientMetadata

Una o più coppie chiave-valore che puoi fornire come input personalizzato al trigger della funzione Lambda del mittente di e-mail personalizzato. Per trasmettere questi dati alla funzione Lambda, usa il parametro ClientMetadata nelle operazioni API AdminRespondToAuthChallenge e RespondToAuthChallenge. Quando trasmette la richiesta alla funzione di post autenticazione, Amazon Cognito non include dati provenienti dal parametro ClientMetadata nelle operazioni API AdminInitiateAuth e InitiateAuth.

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

Parametri di risposta del mittente di e-mail personalizzato

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta del mittente di e-mail personalizzato. La funzione può utilizzare le operazioni API per interrogare e modificare le risorse o registrare i metadati degli eventi in un sistema esterno.

Attivazione del trigger Lambda del mittente di e-mail personalizzato

Per impostare un trigger del mittente di e-mail personalizzato che utilizza la logica personalizzata per inviare messaggi e-mail per il bacino d'utenza, attiva il trigger come indicato di seguito. La procedura che segue assegna un trigger e-mail personalizzato, un trigger SMS personalizzato o entrambi al pool di utenti. Dopo che hai aggiunto il trigger del mittente e-mail personalizzato, Amazon Cognito invia sempre gli attributi utente, incluso l'indirizzo e-mail, e il codice di verifica alla funzione Lambda quando avrebbe altrimenti inviato un messaggio e-mail con Amazon Simple Email Service.

> **Important**
>
> Amazon Cognito crea una sequenza di escape HTML di caratteri riservati come < (`<`) e > (`>`) nella password temporanea dell'utente. Questi caratteri potrebbero essere visualizzati nelle password temporanee inviate da Amazon Cognito alla funzione del mittente e-mail personalizzata, ma non vengono visualizzati nei codici di verifica temporanei. Per inviare password temporanee, la funzione Lambda deve eliminare questi caratteri dopo aver decrittografato la password e prima di inviare il messaggio all'utente.

1. Creare una chiave di crittografia in AWS KMS. Questa chiave cripta le password temporanee e i codici di autorizzazione generati da Amazon Cognito. Puoi quindi decrittografare questi segreti nella funzione Lambda del mittente personalizzato e inviarli all'utente come testo non crittografato.
2. Concedi al principale del servizio Amazon Cognito l'accesso `cognito-idp.amazonaws.com` per crittografare i codici con la chiave KMS.

   Applica la seguente policy basata sulle risorse alla chiave KMS.

```
{
    "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. Crea una funzione Lambda per il trigger del mittente personalizzato. Amazon Cognito utilizza l'SDK di crittografia AWS per crittografare i segreti, le password temporanee e i codici di autorizzazione delle richieste API degli utenti.

   - Assegna alla funzione Lambda un ruolo IAM che disponga, almeno, delle autorizzazioni `kms:Decrypt` per la tua chiave KMS.

4. Concedi l'accesso `cognito-idp.amazonaws.com` al principale del servizio Amazon Cognito per richiamare la funzione Lambda.

   Il comando della AWS CLI seguente concede ad Amazon Cognito l'autorizzazione per invocare la funzione Lambda:

   ```
   aws lambda add-permission --function-name lambda_arn --statement-id
    "CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
   idp.amazonaws.com
   ```

5. Componi il codice della funzione Lambda per inviare i tuoi messaggi. Amazon Cognito utilizza AWS Encryption SDK per crittografare i segreti prima che vengano inviati da Amazon Cognito alla funzione Lambda del mittente personalizzato. Nella tua funzione, esegui la decrittografia del segreto ed elabora tutti i metadati pertinenti. Quindi invia il codice, il messaggio personalizzato e il numero di telefono di destinazione all'API personalizzata che distribuisce il messaggio.
6. Aggiungi AWS Encryption SDK alla tua funzione Lambda. Per ulteriori informazioni, consulta la sezione relativa ai linguaggi di programmazione di SDK di crittografia AWS. Per aggiornare il pacchetto Lambda, completa la procedura seguente.

   a. Esporta la funzione Lambda come file .zip in AWS Management Console.
   b. Apri la funzione e aggiungi AWS Encryption SDK. Per ulteriori informazioni e i link di download, consulta la sezione relative ai linguaggi di programmazione di AWS Encryption SDK nella Guida per gli sviluppatori di AWS Encryption SDK.
   c. Comprimi la funzione con le dipendenze dell'SDK e carica la funzione in Lambda. Per ulteriori informazioni, consulta Distribuzione di funzioni Lambda come archivi di file .zip nella Guida per gli sviluppatori di AWS Lambda.

7. Aggiorna il pool di utenti per aggiungere trigger Lambda del mittente personalizzati. Includi un parametro `CustomSMSSender` o `CustomEmailSender` in una richiesta API `UpdateUserPool`. L'operazione API `UpdateUserPool` richiede tutti i parametri del pool di utenti e i parametri che desideri modificare. Se non fornisci tutti i parametri rilevanti, Amazon Cognito imposta i valori di

tutti i parametri mancanti sui valori predefiniti. Come illustrato nell'esempio che segue, includi le voci per tutte le funzioni Lambda che desideri aggiungere o mantenere nel tuo pool di utenti. Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti.

```
   #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"
```

Per rimuovere un trigger Lambda del mittente personalizzato con una `update-user-pool` AWS CLI, ometti il parametro `CustomSMSSender` o `CustomEmailSender` da `--lambda-config` e includi tutti gli altri trigger che desideri utilizzare con il pool di utenti.

Per rimuovere un trigger Lambda del mittente personalizzato con una richiesta API `UpdateUserPool`, ometti il parametro `CustomSMSSender` o `CustomEmailSender` dal corpo della richiesta contenente il resto della configurazione del pool di utenti.

Esempio di codice

Nel seguente esempio Node.js viene illustrato come elaborare un evento di messaggio e-mail nella funzione Lambda del mittente di e-mail personalizzata. Per questo esempio si presuppone che la funzione abbia due variabili d'ambiente definite.

`KEY_ALIAS`

L'alias della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

`KEY_ARN`

Il nome della risorsa Amazon (ARN) della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

```
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;
};
```

Origini dei trigger Lambda del mittente di e-mail personalizzato

La tabella che segue mostra gli eventi di attivazione per le origini dei trigger delle e-mail personalizzate nel codice Lambda.

TriggerSource value	Evento
CustomEmailSender_SignUp	Un utente si registra e Amazon Cognito invia un messaggio di benvenuto.
CustomEmailSender_ForgotPassword	Un utente richiede un codice per reimpostare la password.
CustomEmailSender_ResendCode	Un utente richiede un codice sostitutivo per reimpostare la password.
CustomEmailSender_UpdateUserAttribute	Un utente aggiorna un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
CustomEmailSender_VerifyUserAttribute	Un utente crea un nuovo attributo indirizzo e-mail o numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
CustomEmailSender_AdminCreateUser	Crei un nuovo utente nel tuo bacino d'utenza e Amazon Cognito gli invia una password temporanea.
CustomEmailSender_AccountTakeOverNotification	Amazon Cognito rileva il tentativo di prendere il controllo di un account utente e invia una notifica all'utente.

Trigger Lambda del mittente di SMS personalizzato

Quando assegni un trigger del mittente SMS personalizzato al pool di utenti, Amazon Cognito richiama una funzione Lambda anziché il suo comportamento predefinito quando un evento utente richiede l'invio di un messaggio e-mail. Con un trigger mittente personalizzato, la tua AWS Lambda funzione può inviare notifiche SMS ai tuoi utenti tramite un metodo e un provider di tua scelta. Il codice personalizzato della funzione deve elaborare e distribuire tutti i messaggi SMS del pool di utenti.

> **Note**
>
> Al momento, non è possibile assegnare trigger del mittente personalizzati nella console Amazon Cognito. Puoi assegnare un trigger con il parametro `LambdaConfig` in una richiesta API `CreateUserPool` o `UpdateUserPool`.

Per usare questo trigger, esegui questi passaggi:

1. Crea una chiave di crittografia simmetrica in AWS Key Management Service ().AWS KMS Amazon Cognito genera segreti (password temporanee, codici di verifica e codici di conferma), quindi utilizza questa chiave KMS per crittografarli. Puoi quindi usare l'operazione API Decrittografa nella funzione Lambda per decrittografare i segreti e inviarli all'utente come testo non crittografato. AWS Encryption SDKÈ uno strumento utile per AWS KMS le operazioni relative alla funzione.
2. Crea una funzione Lambda che desideri assegnare come trigger del mittente personalizzato. Concedi al ruolo della funzione Lambda autorizzazioni `kms:Decrypt` per la chiave KMS.
3. Concedi l'accesso `cognito-idp.amazonaws.com` al principale del servizio Amazon Cognito per richiamare la funzione Lambda.
4. Scrivi il codice della funzione Lambda che indirizza i messaggi ai metodi di distribuzione personalizzati o a fornitori di terze parti. Per distribuire il codice di verifica o conferma dell'utente, Base64 decodifica e decrittografa il valore del parametro `code` nella richiesta. Questa operazione produce un codice o una password in testo normale da includere nel messaggio.
5. Aggiorna il bacino d'utenza in modo che utilizzi un trigger Lambda del mittente personalizzato. Il principale IAM che aggiorna o crea un pool di utenti con un trigger del mittente personalizzato deve disporre dell'autorizzazione per creare una concessione per la tua chiave KMS. Il seguente frammento `LambdaConfig` assegna funzioni del mittente SMS ed e-mail personalizzate.

```
"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"
```

```
    }
```

Parametri del trigger Lambda del mittente di SMS personalizzato

La richiesta passata da Amazon Cognito a questa funzione Lambda è una combinazione dei parametri seguenti e dei parametri comuni aggiunti da Amazon Cognito a tutte le richieste.

JSON

```
{
    "request": {
        "type": "customSMSSenderRequestV1",
        "code": "string",
        "clientMetadata": {
            "string": "string",
             . . .
            },
        "userAttributes": {
            "string": "string",
            . . .
         }
}
```

Parametri di richiesta del mittente di SMS personalizzato

type (tipo)

La versione della richiesta. Per un evento del mittente di SMS personalizzato, il valore di questa stringa è sempre `customSMSSenderRequestV1`.

code

Il codice crittografato che la funzione può decrittografare e inviare all'utente.

clientMetadata

Una o più coppie chiave-valore che è possibile fornire come input personalizzato al trigger della funzione Lambda del mittente di SMS personalizzato. Per passare questi dati alla funzione Lambda, puoi utilizzare il ClientMetadata parametro nelle azioni AdminRespondToAuthChallengee RespondToAuthChallengeAPI. Amazon Cognito non include i dati del ClientMetadata parametro

AdminInitiateAuthe le operazioni InitiateAuthAPI nella richiesta che passa alla funzione di post-autenticazione.

userAttributes

Una o più coppie chiave-valore che rappresentano gli attributi utente.

Parametri di risposta del mittente di SMS personalizzato

Amazon Cognito non prevede di restituire ulteriori informazioni nella risposta. La funzione può utilizzare le operazioni API per interrogare e modificare le risorse o registrare i metadati degli eventi in un sistema esterno.

Attivazione del trigger Lambda del mittente di SMS personalizzato

Puoi impostare un trigger del mittente di SMS personalizzato che utilizza la logica personalizzata per inviare messaggi SMS per il tuo pool di utenti. La procedura seguente assegna un trigger SMS personalizzato, un trigger e-mail personalizzato o entrambi al pool di utenti. Dopo che hai aggiunto il trigger del mittente di SMS personalizzato, Amazon Cognito invia sempre gli attributi utente, incluso il numero di telefono e il codice di verifica, alla funzione Lambda al posto del comportamento predefinito che prevede l'invio di un messaggio SMS con Amazon Simple Notification Service.

> **Important**
>
> Amazon Cognito crea una sequenza di escape HTML di caratteri riservati come < (`<`) e > (`>`) nella password temporanea dell'utente. Questi caratteri potrebbero essere visualizzati nelle password temporanee inviate da Amazon Cognito alla funzione del mittente e-mail personalizzata, ma non vengono visualizzati nei codici di verifica temporanei. Per inviare password temporanee, la funzione Lambda deve eliminare questi caratteri dopo aver decrittografato la password e prima di inviare il messaggio all'utente.

1. Creare una chiave di crittografia in AWS KMS. Questa chiave cripta le password temporanee e i codici di autorizzazione generati da Amazon Cognito. Puoi quindi decrittografare questi segreti nella funzione Lambda del mittente personalizzato e inviarli all'utente come testo non crittografato.
2. Concedi al principale del servizio Amazon Cognito l'accesso `cognito-idp.amazonaws.com` per crittografare i codici con la chiave KMS.

Applica la seguente policy basata sulle risorse alla chiave KMS.

```
{
    "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. Crea una funzione Lambda per il trigger del mittente personalizzato. Amazon Cognito utilizza l'SDK di crittografia AWS per crittografare i segreti, le password temporanee e i codici di autorizzazione delle richieste API degli utenti.

   - Assegna alla funzione Lambda un ruolo IAM che disponga, almeno, delle autorizzazioni `kms:Decrypt` per la tua chiave KMS.

4. Concedi l'accesso `cognito-idp.amazonaws.com` al principale del servizio Amazon Cognito per richiamare la funzione Lambda.

   Il AWS CLI comando seguente concede ad Amazon Cognito l'autorizzazione a richiamare la funzione Lambda:

```
        aws lambda add-permission --function-name lambda_arn --statement-id
 "CognitoLambdaInvokeAccess" --action lambda:InvokeFunction --principal cognito-
idp.amazonaws.com
```

5. Componi il codice della funzione Lambda per inviare i tuoi messaggi. Amazon Cognito utilizza AWS Encryption SDK la crittografia dei segreti prima che Amazon Cognito li invii alla funzione Lambda personalizzata del mittente. Nella tua funzione, esegui la decrittografia del segreto ed elabora tutti i metadati pertinenti. Quindi invia il codice, il messaggio personalizzato e il numero di telefono di destinazione all'API personalizzata che distribuisce il messaggio.
6. Aggiungi il AWS Encryption SDK alla tua funzione Lambda. Per ulteriori informazioni, consulta la sezione relativa ai linguaggi di programmazione di SDK di crittografia AWS. Per aggiornare il pacchetto Lambda, completa la procedura seguente.

   a. Esporta la funzione Lambda come file .zip in AWS Management Console.

   b. Apri la tua funzione e aggiungi il AWS Encryption SDK. Per ulteriori informazioni e i link di download, consulta la sezione relative ai linguaggi di programmazione di AWS Encryption SDK nella Guida per gli sviluppatori di AWS Encryption SDK .

   c. Comprimi la funzione con le dipendenze dell'SDK e carica la funzione in Lambda. Per ulteriori informazioni, consulta Distribuzione di funzioni Lambda come archivi di file .zip nella Guida per gli sviluppatori di AWS Lambda .
7. Aggiorna il pool di utenti per aggiungere trigger Lambda del mittente personalizzati. Includi un parametro `CustomSMSSender` o `CustomEmailSender` in una richiesta API `UpdateUserPool`. L'operazione API `UpdateUserPool` richiede tutti i parametri del pool di utenti e i parametri che desideri modificare. Se non fornisci tutti i parametri rilevanti, Amazon Cognito imposta i valori di tutti i parametri mancanti sui valori predefiniti. Come illustrato nell'esempio che segue, includi le voci per tutte le funzioni Lambda che desideri aggiungere o mantenere nel tuo pool di utenti. Per ulteriori informazioni, consulta Aggiornamento della configurazione del pool di utenti.

```
   #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"
```

Per rimuovere un trigger Lambda del mittente personalizzato con `update-user-pool` AWS CLI un, ometti `CustomSMSSender` il parametro `CustomEmailSender` o e `--lambda-config` includi tutti gli altri trigger che desideri utilizzare con il tuo pool di utenti.

Per rimuovere un trigger Lambda del mittente personalizzato con una richiesta API `UpdateUserPool`, ometti il parametro `CustomSMSSender` o `CustomEmailSender` dal corpo della richiesta contenente il resto della configurazione del pool di utenti.

esempio di codice

Nel seguente esempio Node.js viene illustrato come elaborare un evento di messaggio SMS nella funzione Lambda del mittente di SMS personalizzata. Per questo esempio si presuppone che la funzione abbia due variabili d'ambiente definite.

`KEY_ALIAS`

L'alias della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

`KEY_ARN`

Il nome della risorsa Amazon (ARN) della chiave KMS che desideri utilizzare per crittografare e decrittografare i codici degli utenti.

```
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;
};
```

Argomenti

- Valutazione delle funzionalità di messaggio SMS con una funzione del mittente di SMS personalizzato
- Origini dei trigger Lambda del mittente di SMS personalizzato

Valutazione delle funzionalità di messaggio SMS con una funzione del mittente di SMS personalizzato

Una funzione Lambda del mittente di SMS personalizzato accetta i messaggi SMS inviati dal bacino d'utenza e fornisce il contenuto in base alla logica personalizzata. Amazon Cognito invia i Parametri del trigger Lambda del mittente di SMS personalizzato alla tua funzione. La funzione può fare ciò che vuoi con queste informazioni. Ad esempio, puoi inviare il codice a un argomento Amazon Simple Notification Service (Amazon SNS). Un sottoscrittore dell'argomento Amazon SNS può essere un messaggio SMS, un endpoint HTTPS o un indirizzo e-mail.

Per creare un ambiente di test per la messaggistica SMS di Amazon Cognito con una funzione Lambda del mittente SMS personalizzata, amazon-cognito-user-poolconsulta development-and-testing-with - sms-redirected-to-email - nella libreria aws-samples su. GitHub Il repository contiene AWS CloudFormation modelli che possono creare un nuovo pool di utenti o utilizzare un pool di utenti che già possiedi. Questi modelli creano funzioni Lambda e un argomento Amazon SNS. La

funzione Lambda che il modello assegna come trigger del mittente di SMS personalizzato, reindirizza i messaggi SMS ai sottoscrittori dell'argomento Amazon SNS.

Quando distribuisci questa soluzione in un bacino d'utenza, tutti i messaggi che Amazon Cognito invia di solito tramite messaggi SMS vengono inviati dalla funzione Lambda a un indirizzo e-mail centrale. Utilizza questa soluzione per personalizzare e visualizzare in anteprima i messaggi SMS e per testare gli eventi del bacino d'utenza che causano l'invio di un messaggio SMS da parte di Amazon Cognito. Dopo aver completato i test, ripristina lo CloudFormation stack o rimuovi l'assegnazione personalizzata della funzione di mittente SMS dal tuo pool di utenti.

> **Important**
>
> Non utilizzate i modelli in amazon-cognito-user-pool- development-and-testing-with - sms-redirected-to-email per creare un ambiente di produzione. La funzione Lambda del mittente di SMS personalizzato nella soluzione simula i messaggi SMS, ma li invia tutti a un unico indirizzo e-mail centrale. Prima di poter inviare messaggi SMS in un bacino d'utenza di Amazon Cognito di produzione, è necessario completare i requisiti indicati in Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

Origini dei trigger Lambda del mittente di SMS personalizzato

La tabella che segue mostra l'evento di attivazione per le origini dei trigger di SMS personalizzati nel codice Lambda.

`TriggerSource value`	Evento
`CustomSMSSender_SignUp`	Un utente si registra e Amazon Cognito invia un messaggio di benvenuto.
`CustomSMSSender_ForgotPassword`	Un utente richiede un codice per reimpostare la password.
`CustomSMSSender_ResendCode`	Un utente richiede un nuovo codice per confermare la registrazione.
`CustomSMSSender_VerifyUserAttribute`	Un utente crea un nuovo attributo indirizzo e-mail o numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.

TriggerSource value	Evento
CustomSMSSender_UpdateUserAttribute	Un utente aggiorna un indirizzo e-mail o un numero di telefono e Amazon Cognito gli invia un codice per verificare l'attributo.
CustomSMSSender_Authentication	Un utente configurato con l'autenticazione a più fattori (MFA) per SMS esegue l'accesso.
CustomSMSSender_AdminCreateUser	Crei un nuovo utente nel tuo bacino d'utenza e Amazon Cognito gli invia una password temporanea.

Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

I pool di utenti di Amazon Cognito sono integrati con Amazon Pinpoint per fornire analisi per i pool di utenti di Amazon Cognito e per migliorare i dati utente per le campagne di Amazon Pinpoint. Amazon Pinpoint fornisce analisi dei dati e campagne mirate per guidare il coinvolgimento degli utenti nelle app per dispositivi mobili tramite notifiche push. Con il supporto di analisi dei dati di Amazon Pinpoint nei bacini d'utenza di Amazon Cognito, puoi tracciare le registrazioni, gli accessi, le autenticazioni non riuscite, gli utenti attivi giornalmente (daily active users, DAU) e utenti attivi mensilmente (monthly active users, MAU) nella console di Amazon Pinpoint. Puoi esplorare i dati in base a intervalli di date o attributi differenti, come piattaforma di dispositivo, dispositivo locale, e versione dell'app.

Puoi inoltre impostare attributi personalizzati per la tua app. Questi possono essere utilizzati per segmentare gli utenti su Amazon Pinpoint e inviare loro notifiche push mirate. Se scegli Share user attribute data with Amazon Pinpoint (Condividi dati attributi utente con Amazon Pinpoint) nella scheda Analytics (Analisi) della console Amazon Cognito, Amazon Pinpoint crea degli endpoint aggiuntivi per gli indirizzi e-mail e i numeri di telefono degli utenti.

Quando attivi le analisi di Amazon Pinpoint nel pool di utenti con la console di Amazon Cognito, crei anche un ruolo collegato ai servizi che Amazon Cognito assume quando effettua una richiesta API ad Amazon Pinpoint per il pool di utenti. Il principale IAM che aggiunge la configurazione di analisi deve disporre delle autorizzazioni CreateServiceLinkedRole. Il ruolo collegato al servizio è AWSServiceRoleForAmazonCognitoIdp. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Quando applichi un `AnalyticsConfiguration` al client dell'app nell'API di Amazon Cognito, puoi assegnare un ruolo IAM personalizzato per Amazon Pinpoint e un ID esterno per assumere il ruolo. Il ruolo deve convalidare il principale del servizio `cognito-idp` e, se la policy di attendibilità del ruolo richiede un ID esterno, deve corrispondere a `AnalyticsConfiguration`. Devi concedere le autorizzazioni `cognito-idp:Describe*` del ruolo e le autorizzazioni seguenti per il progetto Amazon Pinpoint.

- `mobiletargeting:UpdateEndpoint`
- `mobiletargeting:PutEvents`

Disponibilità nelle regioni Amazon Cognito e Amazon Pinpoint

Nella tabella riportata di seguito vengono mostrate le mappature Regione AWS tra Amazon Cognito e Amazon Pinpoint che soddisfano una delle condizioni seguenti.

- È possibile utilizzare un progetto Amazon Pinpoint solo nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1).
- È possibile utilizzare un progetto Amazon Pinpoint nella stessa regione o nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1)

Per impostazione predefinita, Amazon Cognito può inviare analisi solo a un progetto Amazon Pinpoint all'interno della stessa Regione AWS. Le eccezioni a questa regola sono le regioni riportate nella tabella seguente e le regioni in cui Amazon Pinpoint non è disponibile.

Amazon Pinpoint è disponibile nelle seguenti regioni. I pool di utenti di Amazon Cognito in queste regioni non supportano l'analisi.

- Europa (Milano)
- Medio Oriente (Bahrein)
- Asia Pacifico (Osaka-Locale)
- Israele (Tel Aviv)
- Africa (Città del Capo)
- Asia Pacifico (Giacarta)

La tabella mostra la relazione tra la regione in cui è stato creato il pool di utenti di Amazon Cognito e la regione corrispondente in Amazon Pinpoint. Il progetto Amazon Pinpoint deve essere configurato in una regione disponibile per integrarlo con Amazon Cognito.

Regione del pool di utenti di Amazon Cognito	Regioni per progetto 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
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

Esempi di mappatura di regioni

- Se crei un pool di utenti in ap-northest-1, puoi creare il progetto Amazon Pinpoint in us-east-1.
- Se crei un pool di utenti in ap-south-1, puoi creare il progetto Amazon Pinpoint in us-east-1 o ap-south-1.

> **Note**
>
> Per tutte le Regioni AWS ad eccezione di quelle nella tabella precedente, Amazon Cognito può utilizzare solo un progetto Amazon Pinpoint nella stessa regione come pool di utenti. Se Amazon Pinpoint non è disponibile nella regione in cui è stato creato il pool di utenti e non è elencato nella tabella, allora Amazon Cognito non supporta l'analisi Amazon Pinpoint in tale regione. Per informazioni sulla Regione AWS dettagliate, consulta Endpoint e quote di Amazon Pinpoint.

Specifica delle impostazioni di analisi dei dati di Amazon Pinpoint (AWS Management Console)

È possibile configurare il pool di utenti di Amazon Cognito per inviare dati di analisi ad Amazon Pinpoint. Amazon Cognito invia dati di analisi solo ad Amazon Pinpoint per utenti locali. Dopo che hai configurato il pool di utenti per l'associazione a un progetto Amazon Pinpoint, devi includere `AnalyticsMetadata` nelle richieste API. Per ulteriori informazioni, consulta Integrazione di app con Amazon Pinpoint.

Specificazione delle impostazioni di analisi

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS.
2. Seleziona User Pools (Pool di utenti) e scegli un pool di utenti esistente dall'elenco.
3. Scegli la scheda App integration (Integrazione app).
4. In App clients and analytics (Client di app e analisi dei dati), scegli un App client name (Nome del client dell'app) esistente dall'elenco.
5. In Pinpoint analytics (Analisi dei dati di Amazon Pinpoint), scegli Enable (Abilita).
6. Scegli una Pinpoint Region (Regione Pinpoint).
7. Scegli un Amazon Pinpoint project (Progetto Amazon Pinpoint) o seleziona Create Amazon Pinpoint project (Crea progetto Amazon Pinpoint).

   > **Note**
   >
   > L'ID progetto di Amazon Pinpoint è una stringa di 32 caratteri univoca per il progetto. È elencato nella console Amazon Pinpoint.

Puoi mappare più app di Amazon Cognito a un singolo progetto di Amazon Pinpoint. Tuttavia, ogni app di Amazon Cognito può essere mappata solo a un progetto di Amazon Pinpoint.
In Amazon Pinpoint, ogni progetto deve essere una singola app. Ad esempio, se uno sviluppatore di videogiochi ha due videogiochi, ognuno di questi dovrebbe costituire un progetto di Amazon Pinpoint separato, nonostante entrambi usino lo stesso bacino d'utenza di Amazon Cognito. Per ulteriori informazioni sui progetti Amazon Pinpoint, consulta Creazione di un progetto in Amazon Pinpoint.

8. In User data sharing (Condivisione dei dati utente), scegli Share user data with Amazon Pinpoint (Condividi i dati utente con Amazon Pinpoint) se Amazon Cognito deve inviare indirizzi e-mail e numeri di telefono ad Amazon Pinpoint e creare endpoint aggiuntivi per gli utenti. Dopo che gli utenti verificano i rispetti indirizzi e-mail e numeri di telefono, Amazon Cognito li condividi solo con Amazon Pinpoint se sono disponibili per l'account utente.

   Note

   Un endpoint identifica in modo univoco un dispositivo dell'utente al quale puoi inviare notifiche push con Amazon Pinpoint. Per ulteriori informazioni sugli endpoint, consulta la sezione Aggiunta di endpoint nella Guida per gli sviluppatori di Amazon Pinpoint.

9. Seleziona Salva modifiche.

Specifica delle impostazioni di analisi dei dati di Amazon Pinpoint (AWS CLI e AWS API)

Utilizza i comandi seguenti per specificare le impostazioni di analisi dei dati di Amazon Pinpoint per il tuo bacino d'utenza.

Specificazione di impostazioni di analisi per l'app client esistente del tuo bacino d'utenza al momento della creazione dell'app.

- AWS CLI: `aws cognito-idp create-user-pool-client`
- API AWS: CreateUserPoolClient

Aggiornamento delle impostazioni di analisi per l'app client esistente del tuo bacino d'utenza

- AWS CLI: `aws cognito-idp update-user-pool-client`
- API AWS: UpdateUserPoolClient

> **Note**
>
> Se utilizzi `ApplicationArn`, Amazon Cognito supporta le integrazioni all'interno della regione.

Integrazione di app con Amazon Pinpoint

Puoi pubblicare metadati di analisi su Amazon Pinpoint per gli utenti locali di Amazon Cognito nell'API dei pool di utenti.

Utenti locali

Sono gli utenti che hanno creato un account mediante la registrazione o sono stati creati nel pool di utenti anziché eseguire l'accesso tramite un gestore dell'identità digitale (IdP) di terze parti.

API dei pool di utenti

Sono le operazioni che puoi integrare con un SDK AWS utilizzando un'app con un'interfaccia utente personalizzata. Non puoi passare metadati di analisi per utenti federati o locali che effettuano l'accesso tramite l'interfaccia utente ospitata. Per un elenco di operazioni dei pool di utenti, consultare la documentazione di riferimento delle API di Amazon Cognito.

Dopo aver configurato il pool di utenti per la pubblicazione in una campagna, Amazon Cognito passa i metadati ad Amazon Pinpoint per le seguenti operazioni API.

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `ConfirmForgotPassword`
- `ConfirmSignUp`
- `ForgotPassword`
- `InitiateAuth`

- `ResendConfirmationCode`
- `RespondToAuthChallenge`
- `SignUp`

Per passare i metadati relativi alla sessione dell'utente alla campagna Amazon Pinpoint, includi un valore `AnalyticsEndpointId` nel parametro `AnalyticsMetadata` della richiesta API. Per un esempio di JavaScript, consulta Perché le analisi del pool di utenti Amazon Cognito non vengono visualizzate nel pannello di controllo di Amazon Pinpoint? nel Portale del sapere AWS.

Configurazione delle analisi di un bacino d'utenza

Utilizzando l'analisi dei dati di Amazon Pinpoint, puoi tenere traccia delle registrazioni, degli accessi, delle autenticazioni non riuscite, degli utenti attivi giornalieri (DAU) e degli utenti attivi mensili (MAU) del bacino d'utenza di Amazon Cognito. Puoi inoltre configurare gli attributi utente specifici per l'app utilizzando AWS Mobile SDK for Android o AWS Mobile SDK for iOS. Questi possono essere utilizzati per segmentare gli utenti su Amazon Pinpoint e inviare loro notifiche push mirate.

Nella scheda Integrazione app in Client di app e analisi dei dati, puoi accedere a un client di app esistente o crearne uno nuovo. Nella configurazione del client di app, puoi specificare un progetto Amazon Pinpoint da utilizzare con la tua app. Per ulteriori informazioni, consulta Utilizzo delle analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

> **Note**
>
> Amazon Pinpoint è disponibile in diverse regioni AWS in Nord America, Europa, Asia e Oceania. Le regioni di Amazon Pinpoint includono l'API Amazon Pinpoint. Se Amazon Cognito supporta una regione Amazon Pinpoint, invierà eventi ai progetti Amazon Pinpoint all'interno della stessa regione Amazon Pinpoint. Se una regione non è supportata da Amazon Pinpoint, Amazon Cognito supporterà solo l'invio di eventi in us-east-1. Per informazioni dettagliate sulla regione di Amazon Pinpoint, consulta Endpoint Amazon Pinpoint e Utilizzo delle analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Aggiunta di analisi e campagne

1. Scegli Add analytics and campaigns (Aggiungi analisi e campagne).
2. Scegli un Cognito app client (Client dell'app di Cognito) dall'elenco.

3. Per mappare la tua app Amazon Cognito a un progetto Amazon Pinpoint, per prima cosa scegli il progetto Amazon Pinpoint dall'elenco.

   > **Note**
   >
   > L'ID progetto di Amazon Pinpoint è una stringa di 32 caratteri univoca per il progetto. È elencato nella console Amazon Pinpoint.
   > Puoi mappare più app di Amazon Cognito a un singolo progetto di Amazon Pinpoint. Tuttavia, ogni app di Amazon Cognito può essere mappata solo a un progetto di Amazon Pinpoint.
   > In Amazon Pinpoint, ogni progetto deve essere una singola app. Ad esempio, se uno sviluppatore di videogiochi ha due videogiochi, ognuno di questi dovrebbe costituire un progetto di Amazon Pinpoint separato anche se entrambi usano lo stesso bacino d'utenza di Amazon Cognito.

4. Seleziona Condividi dati attributi utente con Amazon Pinpoint se desideri che Amazon Cognito invii gli indirizzi e-mail e i numeri di telefono a Amazon Pinpoint al fine di creare endpoint agguntivi per gli utenti.

   > **Note**
   >
   > Un endpoint identifica in modo univoco un dispositivo dell'utente al quale puoi inviare notifiche push con Amazon Pinpoint. Per ulteriori informazioni sugli endpoint, consulta la sezione Aggiunta di endpoint ad Amazon Pinpoint nella Guida per gli sviluppatori di Amazon Pinpoint.

5. Inserisci un Ruolo IAM che hai già creato oppure seleziona Crea nuovo ruolo per creare un nuovo ruolo nella console IAM.
6. Scegli Save changes (Salva modifiche).
7. Per specificare mappature aggiuntive dell'app, seleziona Add app mapping (Aggiungi mappatura dell'app).
8. Scegli Save changes (Salva modifiche).

Gestione degli utenti nel tuo bacino d'utenza

Dopo che hai creato un pool di utenti, puoi creare, confermare e gestire gli account utente. Con i gruppi di bacini d'utenza di Amazon Cognito, è possibile gestire gli utenti e il loro accesso alle risorse mappando i ruoli IAM ai gruppi.

È possibile importare gli utenti in un bacino d'utenza con un trigger Lambda di migrazione utenti. Questo approccio consente di semplificare la migrazione di utenti dalla directory utente esistente ai bacini d'utenza quando accedono per la prima volta al bacino d'utenza.

Argomenti

- Configurazione delle policy per la creazione degli utenti
- Registrazione e conferma degli account utente
- Creazione di account utente come amministratore
- Aggiunta di gruppi a un bacino d'utenza
- Gestione e ricerca degli account utente
- Recupero degli account utente
- Importazione di utenti in un bacino d'utenza
- Attributi del bacino d'utenza
- Aggiunta di requisiti password del bacino d'utenza

Configurazione delle policy per la creazione degli utenti

Il pool di utenti può consentire agli utenti di registrarsi oppure puoi crearli come amministratore. Puoi anche controllare in che misura il processo di verifica e conferma dopo la registrazione è affidato agli utenti. Ad esempio, potresti voler esaminare le iscrizioni e accettarle sulla base di un processo di convalida esterno. Questa configurazione, o policy utente creata dall'amministratore, imposta inoltre il periodo di tempo prima che un utente non possa più confermare il proprio account.

Amazon Cognito può soddisfare le esigenze dei clienti pubblici come piattaforma CIAM (Customer Identity and Access Management) per il software. Un pool di utenti che accetta l'iscrizione e dispone di un client per l'app, con o senza un'interfaccia utente ospitata, crea un profilo utente per chiunque su Internet conosca l'ID client dell'app, individuabile pubblicamente, e richieda di registrarsi. Un profilo utente registrato può ricevere token di accesso e di identità e può accedere alle risorse che hai autorizzato per l'app. Prima di attivare l'iscrizione nel pool di utenti, esamina le opzioni e assicurati

che la configurazione sia conforme ai tuoi standard di sicurezza. Imposta Abilita la registrazione self-service e `AllowAdminCreateUserOnly`, come descritto nelle seguenti procedure, con attenzione.

AWS Management Console

La scheda Esperienza di registrazione del pool di utenti e la fase Configurazione dell'esperienza di iscrizione della procedura guidata per la creazione del pool di utenti contengono alcune impostazioni per la registrazione e la creazione amministrativa degli utenti nel pool di utenti.

Configurazione dell'esperienza di registrazione

1. In Verifica e conferma assistite da Cognito, scegli se Consentire a Cognito di inviare automaticamente messaggi per la verifica e la conferma. Con questa impostazione abilitata, Amazon Cognito invia un'e-mail o un messaggio SMS ai nuovi utenti con un codice da presentare al pool di utenti. Ciò conferma la proprietà dell'indirizzo e-mail o del numero di telefono, impostando l'attributo equivalente come verificato e confermando l'account utente per l'accesso. Gli Attributi da verificare scelti determinano i metodi di consegna e le destinazioni dei messaggi di verifica.
2. Verifica delle modifiche degli attributi non è importante quando si creano utenti, ma è relativa alla verifica degli attributi. Puoi consentire agli utenti che hanno modificato ma non ancora verificato i propri attributi di accesso di continuare ad accedere con il nuovo valore dell'attributo o con quello originale. Per ulteriori informazioni, consulta Verifica in caso di modifica dell'e-mail o del numero di telefono da parte dell'utente.
3. Attributi obbligatori mostra gli attributi a cui è necessario fornire un valore prima che un utente possa registrarsi o creare un utente. Puoi impostare gli attributi obbligatori solo nella procedura guidata per la creazione del pool di utenti.
4. Gli Attributi personalizzati sono importanti per il processo di creazione e registrazione degli utenti perché puoi impostare un valore per gli attributi personalizzati non modificabili solo quando crei un utente per la prima volta. Per ulteriori informazioni sugli attributi personalizzati, consulta Attributi personalizzati.
5. In Iscrizione self-service seleziona Abilita la registrazione self-service se desideri che gli utenti siano in grado di generare un nuovo account con l'API `SignUp` non autenticata. Se disabiliti la registrazione self-service, puoi creare nuovi utenti solo come amministratore, nella console Amazon Cognito o con richieste API AdminCreateUser. In un pool di utenti in cui la registrazione self-service è inattiva, le richieste dell'API SignUp restituiscono `NotAuthorizedException` e l'interfaccia utente ospitata non mostra un link di Registrazione.

Per i pool di utenti in cui prevedi di creare utenti come amministratore, puoi configurare la durata delle loro password temporanee nella scheda Esperienza di accesso in Scadenza delle password temporanee impostate dagli amministratori.

Un altro elemento importante della creazione di utenti come amministratore è il messaggio di invito. Quando crei un nuovo utente, Amazon Cognito invia un messaggio con un link alla tua app in modo che l'utente possa accedere per la prima volta. Personalizza questo modello di messaggio nella scheda Messaggistica in Modelli di messaggio.

Puoi configurare client di app riservati, in genere applicazioni Web, con un segreto del client che impedisce l'iscrizione senza il segreto del client dell'app. Come best practice consigliata in materia di sicurezza, non distribuire i segreti dei client delle app in client di app pubbliche, in genere app per dispositivi mobili. Puoi creare client di app con segreti del client nella scheda Integrazione app della console Amazon Cognito.

Amazon Cognito user pools API

Puoi impostare a livello di codice i parametri per la creazione di utenti in un pool di utenti in una richiesta API CreateUserPool o UpdateUserPool.

L'elemento AdminCreateUserConfig imposta i valori per le seguenti proprietà di un pool di utenti.

1. Abilitazione dell'iscrizione self-service
2. Il messaggio di invito che invii ai nuovi utenti creati dall'amministratore

L'esempio seguente, quando viene aggiunto al corpo completo della richiesta API, imposta un pool di utenti con l'iscrizione self-service inattiva e un'e-mail di invito di 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
 {####}."
      }
   }
```

I seguenti parametri aggiuntivi di una richiesta API CreateUserPool o UpdateUserPool regolano la creazione di nuovi utenti.

AutoVerifiedAttributes

Gli attributi, gli indirizzi e-mail o i numeri di telefono a cui desideri inviare automaticamente un messaggio quando registri un nuovo utente.

Policy

La policy delle password del pool di utenti.

Schema

Gli attributi personalizzati del pool di utenti. Sono importanti per il processo di creazione e registrazione degli utenti perché puoi impostare un valore per gli attributi personalizzati non modificabili solo quando crei un utente per la prima volta.

Questo parametro imposta anche gli attributi richiesti per il pool di utenti. Il testo seguente, quando viene inserito nell'elemento `Schema` di un corpo di una richiesta API completa, imposta l'attributo `email`come richiesto.

```
{
            "Name": "email",
            "Required": true
}
```

Registrazione e conferma degli account utente

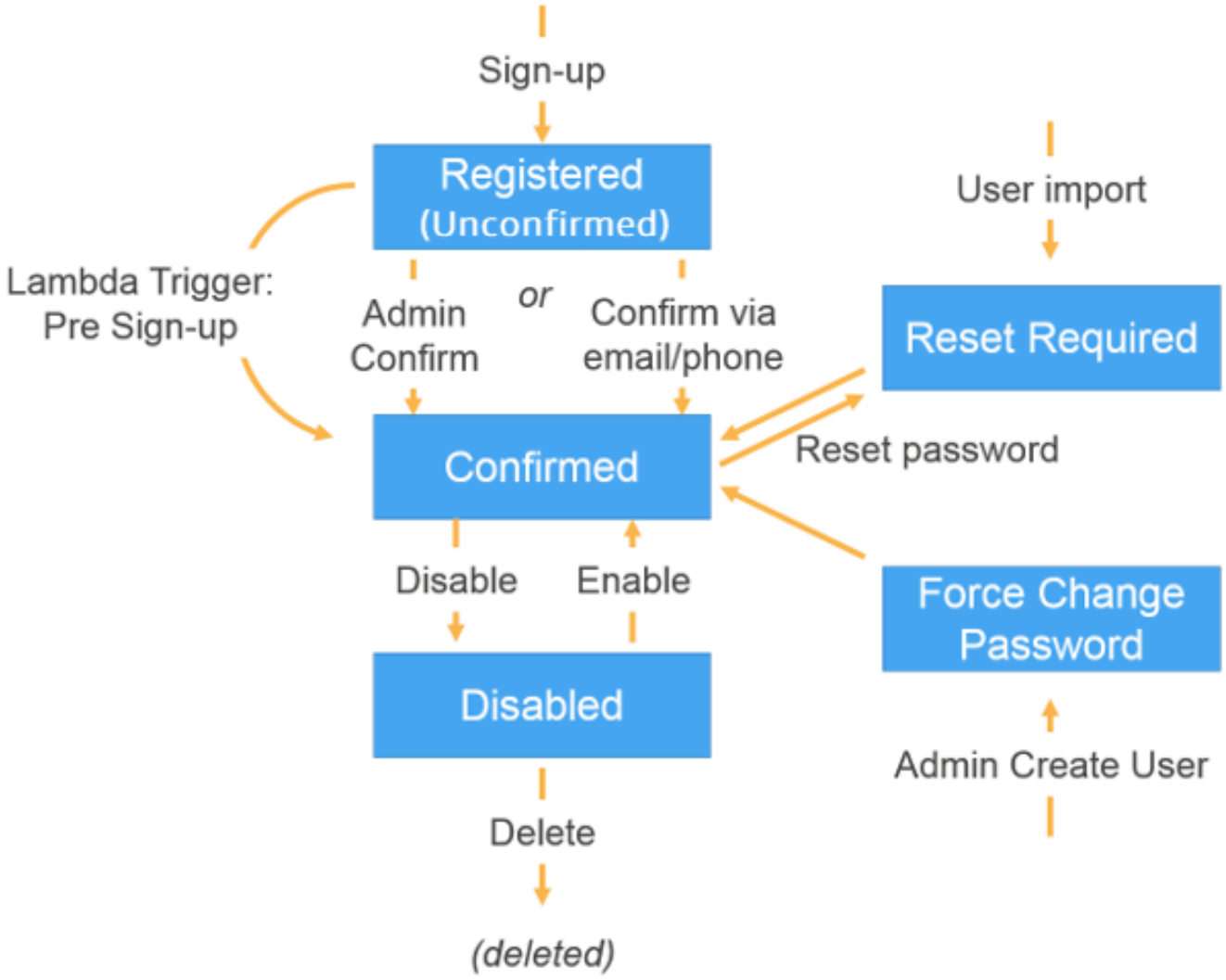
Gli account utente vengono aggiunti al bacino d'utenza in uno dei seguenti modi:

- L'utente effettua la registrazione nell'app client del tuo bacino d'utenza. Può trattarsi di un'app per dispositivi mobili o Web.
- Puoi importare l'account utente nel tuo bacino d'utenza. Per ulteriori informazioni, consulta Importazione di utenti nel bacino d'utenza da un file CSV.
- Puoi creare l'account utente all'interno del bacino d'utenza e invitare l'utente a effettuare l'accesso. Per ulteriori informazioni, consulta Creazione di account utente come amministratore.

Gli utenti che effettuano la registrazione devono essere confermati prima di poter eseguire l'accesso. Gli utenti importati e creati sono già confermati, ma devono creare la propria password quando effettuano l'accesso per la prima volta. Le sezioni seguenti spiegano il processo di conferma e la verifica tramite e-mail o telefono.

Panoramica sulla conferma dell'account utente

Il seguente diagramma illustra il processo di conferma:



Un account utente può avere differenti stati:

Registrato (non confermato)

L'utente si è registrato correttamente, ma non può effettuare l'accesso fino a quando l'account utente non viene confermato. In questo stato l'utente è abilitato ma non confermato.

I nuovi utenti che effettuano la registrazione iniziano da questo stato.

Confermato

L'account utente viene confermato e l'utente può effettuare l'accesso. Quando un utente inserisce un codice o segue un link e-mail per confermare il proprio account utente, tale e-mail o numero di telefono viene verificato automaticamente. Il codice o il link è valido 24 ore.

Se l'account utente è stato confermato dall'amministratore o da un trigger Lambda di pre-registrazione, potrebbe non esserci un'e-mail verificata o un numero di telefono associato all'account.

Reimpostazione della password obbligatoria

L'account utente viene confermato, ma l'utente deve richiedere un codice e reimpostare la password prima di poter effettuare l'accesso.

Gli account utente importati da un amministratore o uno sviluppatore iniziano da questo stato.

Modifica forzata della password

L'account utente viene verificato e l'utente può accedere utilizzando una password temporanea che dovrà modificare al primo accesso con un nuovo valore prima di fare qualsiasi altra operazione.

Gli account utente creati da un amministratore o uno sviluppatore iniziano da questo stato.

Disabilitato

Prima di poter eliminare un account utente, è necessario disabilitare l'accesso per quest'ultimo.

Verifica delle informazioni di contatto al momento della registrazione

Quando nuovi utenti effettuano la registrazione nella tua app, è probabile che tu voglia che forniscano almeno un metodo di contatto. Ad esempio, con le informazioni di contatto degli utenti, puoi:

- Inviare una password temporanea quando un utente sceglie di reimpostare la propria password.
- Avvisare gli utenti quando le loro informazioni personali o finanziarie vengono aggiornate.
- Inviare messaggi promozionali, ad esempio offerte speciali o sconti.
- Inviare riepiloghi sugli account o promemoria di fatturazione.

Per i casi d'uso come questi, è importante inviare messaggi a una destinazione verificata. Diversamente potresti inviare i tuoi messaggi a indirizzi e-mail o numeri di telefono non validi non

digitati correttamente. Ancora peggio, potresti inviare informazioni riservate a destinatari scaltri che fingono di essere tuoi utenti.

Per essere certo di inviare i messaggi solo alle persone corrette, configura il tuo bacino d'utenza Amazon Cognito in modo che gli utenti debbano fornire le informazioni seguenti al momento della registrazione:

a. Un indirizzo e-mail o numero di telefono.
b. Un codice di verifica che Amazon Cognito invia all'indirizzo e-mail o al numero di telefono. Se sono trascorse 24 ore e il codice o il link dell'utente non è più valido, chiama l'operazione ResendConfirmationCodeAPI per generare e inviare un nuovo codice o collegamento.

Fornendo il codice di verifica, un utente dimostra di avere accesso alla mailbox o al telefono che ha ricevuto il codice. Dopo che l'utente ha fornito il codice, Amazon Cognito aggiorna le informazioni relative all'utente nel bacino d'utenza nel seguente modo:

- Impostando lo stato dell'utente su `CONFIRMED`.
- Aggiornando gli attributi dell'utente per indicare che l'indirizzo e-mail o il numero di telefono sono verificati.

Per visualizzare queste informazioni, puoi utilizzare la console Amazon Cognito. In alternativa, puoi utilizzare l'operazione `AdminGetUser` API, il `admin-get-user` comando con o un'azione corrispondente in uno degli AWS SDK. AWS CLI

Se un utente dispone di un metodo di contatto verificato, Amazon Cognito gli invia automaticamente un messaggio quando richiede un ripristino della password.

Per configurare il bacino d'utenza per richiedere la verifica di e-mail o telefono

Quando verifichi gli indirizzi e-mail e i numeri di telefono degli utenti, assicurati di poter contattare questi ultimi. Completa i seguenti passaggi AWS Management Console per configurare il tuo pool di utenti in modo che gli utenti confermino i propri indirizzi e-mail o numeri di telefono.

> **Note**
>
> Se non disponi ancora di un pool di utenti nel tuo account, consulta Nozioni di base sui bacini d'utenza.

Per configurare il bacino d'utenza

1. Passa alla console di Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Dal riquadro di navigazione, scegli User Pools (Bacini d'utenza). Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Sign-up experience (Esperienza di registrazione) e individua l'opzione Attribute verification and user account confirmation (Verifica degli attributi e conferma dell'account utente). Scegli Modifica.
4. In Verifica e conferma assistita da Cognito, se desideri, scegli Consenti a Cognito di inviare automaticamente messaggi per la verifica e la conferma. Con questa impostazione abilitata, Amazon Cognito invia messaggi agli attributi di contatto utente selezionati quando un utente si iscrive o quando crei un profilo utente. Per verificare gli attributi e confermare i profili utente per l'accesso, Amazon Cognito invia un codice o un link nei messaggi agli utenti. Gli utenti devono quindi inserire il codice nell'interfaccia utente in modo che l'app possa confermarli in una richiesta API `ConfirmSignUp` o `AdminConfirmSignUp`.

> **Note**
>
> Puoi anche disabilitare la verifica e conferma assistita da Cognito e utilizzare azioni API autenticate o trigger Lambda per verificare gli attributi e confermare gli utenti. Scegliendo questa opzione, Amazon Cognito non invia alcun codice di verifica quando gli utenti effettuano la registrazione. Scegli questa opzione se utilizzi un flusso di autenticazione personalizzato che verifica almeno un metodo di contatto senza usare i codici di verifica di Amazon Cognito. Ad esempio, è possibile utilizzare un trigger Lambda di pre-registrazione che verifica automaticamente gli indirizzi e-mail che appartengono a un determinato dominio.
> Se non vengono verificate le informazioni di contatto degli utenti, in alcuni casi essi potrebbero non essere in grado di utilizzare l'app. Si noti che tutti gli utenti necessitano di informazioni di contatto verificate per:
>
> - Reimpostare la password utente — Quando un utente effettua un'operazione nell'app che chiama l'operazione API `ForgotPassword`, Amazon Cognito invia una password temporanea all'indirizzo e-mail o al numero di telefono dell'utente. Amazon Cognito invia questa password solo se l'utente ha almeno un metodo di contatto verificato.
> - Accesso tramite indirizzo e-mail o numero di telefono come alias — Configurando il bacino d'utenza per consentire questi alias, un utente sarà in grado di accedere con

un alias solo se è verificato. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

5. Scegli gli attributi da verificare:

   Invia un messaggio SMS, verifica il numero di telefono

   Amazon Cognito invia un codice di verifica tramite messaggio SMS all'accesso dell'utente. Scegli questa opzione se in genere comunichi con gli utenti tramite SMS. Ad esempio, si vorranno utilizzare numeri di telefono verificati se si inviano notifiche di consegna, conferme di appuntamenti o avvisi. I numeri di telefono dell'utente saranno l'attributo verificato al momento della conferma degli account; è necessario intraprendere ulteriori azioni per verificare e comunicare con gli indirizzi e-mail dell'utente.

   Invia messaggio e-mail, verifica l'indirizzo email

   Amazon Cognito invia un codice di verifica tramite messaggio SMS all'accesso dell'utente. Scegliere questa opzione se in genere si comunica con gli utenti tramite e-mail. Ad esempio, si vorranno utilizzare indirizzi e-mail verificati se si inviano estratti conto, riepiloghi di ordini o offerte speciali. Gli indirizzi e-mail dell'utente saranno l'attributo verificato al momento della conferma degli account; è necessario intraprendere ulteriori azioni per verificare e comunicare con gli indirizzi e-mail dell'utente.

   Invia messaggio SMS se il numero di telefono è disponibile, altrimenti invia un messaggio e-mail

   Scegliere questa opzione se non si necessita che tutti gli utenti dispongano del medesimo metodo di contatto verificato. In questo caso, la pagina di registrazione nell'app potrebbe richiedere agli utenti di verificare solo il metodo di contatto preferito. Quando Amazon Cognito un codice di verifica, lo invia al metodo di contatto fornito nella richiesta `SignUp` dall'app. Se un utente fornisce sia un indirizzo e-mail che un numero di telefono e l'app offre entrambi i metodi di contatto nella richiesta `SignUp`, Amazon Cognito invierà il codice di verifica solo al numero di telefono.

   Se occorre che gli utenti verifichino sia un indirizzo e-mail che un numero di telefono, scegliere questa opzione. Amazon Cognito verifica un metodo di contatto quando l'utente effettua la registrazione e l'app deve verificare l'altro metodo di contatto dopo che l'utente ha effettuato l'accesso. Per ulteriori informazioni, consulta Se si richiede agli utenti di confermare sia gli indirizzi e-mail che i numeri di telefono.

6. Scegli Salva modifiche.

Flusso di autenticazione con verifica di e-mail o telefono

Se il bacino d'utenza richiede agli utenti di verificare le proprie informazioni di contatto, l'app deve facilitare il seguente flusso quando un utente effettua la registrazione:

1. L'utente accede all'app inserendo un nome utente, un numero di telefono e/o un indirizzo e-mail ed eventuali altri attributi.
2. Il servizio Amazon Cognito riceve la richiesta di registrazione dall'app. Dopo aver verificato che la richiesta contenga tutti gli attributi necessari per la registrazione, il servizio completa il processo di registrazione e invia un codice di conferma al telefono (tramite SMS) o all'indirizzo e-mail dell'utente. Il codice è valido 24 ore.
3. Il servizio informa l'app che la registrazione è stata completata e che l'account utente è in attesa di conferma. La risposta contiene informazioni su dove è stato inviato il codice di conferma. A questo punto l'account utente è in stato non confermato e l'indirizzo e-mail e il numero di telefono dell'utente non sono verificati.
4. L'app è ora in grado di richiedere all'utente di inserire il codice di conferma. Non è necessario che l'utente inserisca il codice immediatamente. Tuttavia, l'utente non sarà in grado di effettuare l'accesso fino a quando non avrà inserito il codice di conferma.
5. L'utente inserisce il codice di conferma nell'app.
6. L'app chiama `ConfirmSignUp` per inviare il codice al servizio Amazon Cognito, il quale verifica il codice e, se è corretto, imposta l'account utente sullo stato confermato. Dopo aver confermato correttamente l'account utente, il servizio Amazon Cognito contrassegna automaticamente l'attributo utilizzato per la conferma (indirizzo e-mail o numero di telefono) come verificato. A meno che il valore di questo attributo non venga modificato, l'utente non dovrà verificarlo nuovamente.
7. A questo punto l'account utente è in stato confermato e l'utente può effettuare l'accesso.

Se si richiede agli utenti di confermare sia gli indirizzi e-mail che i numeri di telefono

Amazon Cognito verifica solo uno dei metodi di contatto quando un utente effettua la registrazione. Nei casi in cui Amazon Cognito deve scegliere tra la verifica di un indirizzo e-mail o di un numero di telefono, sceglie di verificare il numero di telefono inviando un codice di verifica tramite SMS. Ad esempio, se configuri il tuo bacino d'utenza per consentire agli utenti di verificare gli indirizzi e-mail o i numeri di telefono e se la tua app fornisce entrambi questi attributi al momento della registrazione, Amazon Cognito verifica solo il numero di telefono. Dopo che un utente ha verificato il proprio numero

di telefono, Amazon Cognito imposta lo stato dell'utente su `CONFIRMED` e all'utente sarà consentito accedere all'app.

Dopo che l'utente ha effettuato l'accesso, la tua app può fornire l'opzione di verifica del metodo di contatto che non è stato verificato durante la registrazione. Per verificare il secondo metodo, la tua app chiama l'operazione API `VerifyUserAttribute`. Tieni presente che questa operazione richiede un parametro `AccessToken` e che Amazon Cognito fornisce solo i token di accesso agli utenti autenticati. Pertanto, puoi verificare il secondo metodo di contatto solo dopo che l'utente ha effettuato l'accesso.

Se hai bisogno che i tuoi utenti verifichino sia gli indirizzi e-mail che i numeri di telefono, effettua quanto segue:

1. Configurare il bacino d'utenza per consentire agli utenti di verificare l'indirizzo e-mail o il numero di telefono.
2. Nel flusso di registrazione dell'app, richiedere agli utenti di fornire un indirizzo e-mail e un numero di telefono. Chiamare l'operazione API `SignUp` e fornire l'indirizzo e-mail e il numero di telefono per il parametro `UserAttributes`. A questo punto, Amazon Cognito invia un codice di verifica al telefono dell'utente.
3. Nell'interfaccia dell'app, presentare una pagina di conferma in cui l'utente possa immettere il codice di verifica. Confermare l'utente chiamando l'operazione API `ConfirmSignUp`. A questo punto, lo stato dell'utente è `CONFIRMED` e il suo numero di telefono verificato, ma non lo è l'indirizzo e-mail.
4. Presentare la pagina di accesso e autenticare l'utente chiamando l'operazione API `InitiateAuth`. Dopo che l'utente è stato autenticato, Amazon Cognito restituisce un token di accesso all'app.
5. Chiama l'operazione API `GetUserAttributeVerificationCode`. Specificare i seguenti parametri nella richiesta.

   - `AccessToken`: il token di accesso restituito da Amazon Cognito quando l'utente ha effettuato l'accesso.
   - `AttributeName`: specifica `"email"` come valore dell'attributo.

   Amazon Cognito invia un codice di verifica all'indirizzo e-mail dell'utente.

6. Presentare una pagina di conferma in cui l'utente possa immettere il codice di verifica. Quando l'utente invia il codice, chiamare l'operazione API `VerifyUserAttribute`. Specificare i seguenti parametri nella richiesta.

   - `AccessToken`: il token di accesso restituito da Amazon Cognito quando l'utente ha effettuato l'accesso.
   - `AttributeName`: specifica `"email"` come valore dell'attributo.
   - `Code`: il codice di verifica fornito dall'utente.

   A questo punto, l'indirizzo e-mail è verificato.

Permettere agli utenti di registrarsi ma confermarli come un amministratore del pool di utenti

Potrebbe essere necessario impedire al pool di utenti di inviare automaticamente messaggi di verifica nel pool di utenti, ma consentire comunque a chiunque di registrarsi per un account. Questo modello lascia spazio, ad esempio, alla revisione umana delle nuove richieste di registrazione e alla convalida ed elaborazione batch delle registrazioni. Puoi confermare nuovi account utente nella console Amazon Cognito o con l'operazione API autenticata tramite IAM. AdminConfirmSignUp Puoi confermare gli account utente come un amministratore a prescindere che il pool di utenti invii o meno messaggi di verifica.

Con questa tecnica, puoi solo confermare l'iscrizione self-service di un utente. Per confermare un utente creato come amministratore, crea una richiesta AdminSetUserPasswordAPI con set to. `Permanent` `True`

1. L'utente accede all'app inserendo un nome utente, un numero di telefono e/o un indirizzo e-mail ed eventuali altri attributi.
2. Il servizio Amazon Cognito riceve la richiesta di registrazione dall'app. Dopo aver verificato che la richiesta contenga tutti gli attributi necessari per la registrazione, il servizio completa il processo di registrazione e indica all'app che la registrazione è stata completata ed è in attesa di conferma. A questo punto l'account utente è in stato non confermato. L'utente non può effettuare l'accesso fino a quando l'account non è confermato.
3. Conferma l'account dell'utente. Devi accedere AWS Management Console o firmare la richiesta API con AWS le credenziali per confermare l'account.

a. Per confermare un utente nella console Amazon Cognito, passa alla scheda Utenti, scegli l'utente che desideri confermare e dal menu Azioni seleziona Conferma.

b. Per confermare un utente nell' AWS API o nella CLI, crea una richiesta AdminConfirmSignUpAPI o admin-confirm-sign-upin. AWS CLI

4. A questo punto l'account utente è in stato confermato e l'utente può effettuare l'accesso.

Calcolo dei valori SecretHash

Assegnare un segreto del cliente al client dell'app riservata come una best practice. Quando si assegna un segreto del client al client dell'app, le richieste API del pool di utenti di Amazon Cognito devono includere un hash che includa il segreto del client nel corpo della richiesta. Per convalidare la conoscenza del segreto del client per le operazioni API nei seguenti elenchi, concatenare il segreto del client con l'ID del client dell'app e il nome dell'utente, quindi codificare in base64 tale stringa.

Quando l'app concede l'accesso agli utenti a un client con un hash segreto, puoi utilizzare il valore di qualsiasi attributo di accesso al pool di utenti come elemento nome utente dell'hash segreto. Quando l'app richiede nuovi token in un'operazione di autenticazione con `REFRESH_TOKEN_AUTH`, il valore dell'elemento nome utente dipende dagli attributi di accesso. Se il pool di utenti non ha `username` come attributo di accesso, imposta il valore hash segreto del valore nome utente della richiesta `sub` dell'utente dal token di accesso o token ID. Quando `username` è un attributo di accesso, imposta il valore hash segreto del nome utente dalla richiesta `username`.

Le seguenti API dei pool di utenti di Amazon Cognito accettano un valore hash del segreto del client in un parametro `SecretHash`.

- ConfirmForgotPassword
- ConfirmSignUp
- ForgotPassword
- ResendConfirmationCode
- SignUp

Inoltre, le seguenti API accettano un valore hash del segreto del client in un parametro `SECRET_HASH`, nei parametri di autenticazione o in una risposta di richiesta di verifica.

Operazione API	Parametro principale per SECRET_HASH
InitiateAuth	AuthParameters
AdminInitiateAuth	AuthParameters
RespondToAuthChallenge	ChallengeResponses
AdminRespondToAuthChallenge	ChallengeResponses

Il valore hash del segreto è un codice di autenticazione dei messaggi mediante algoritmi hash con chiave (HMAC) codificato in Base64 e calcolato utilizzando la chiave segreta del client di un pool di utenti e nome utente, oltre all'ID client nel messaggio. Il seguente pseudocodice mostra come viene calcolato questo valore. In questo pseudocodice, + indica la concatenazione, `HMAC_SHA256` rappresenta una funzione che produce un valore HMAC utilizzando HmacSHA256 e `Base64` rappresenta una funzione che genera una versione con codificazione Base-64 dell'output hash.

```
Base64 ( HMAC_SHA256 ( "Client Secret Key", "Username" + "Client Id" ) )
```

Per una panoramica dettagliata su come calcolare e utilizzare il `SecretHash` parametro, consulta Come posso risolvere gli errori «Impossibile verificare l'hash segreto per il client» dall'API dei miei pool di utenti di Amazon Cognito<client-id>? nel Knowledge Center. AWS

Puoi utilizzare i seguenti esempi di codice nel codice dell'app lato server.

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)
```

Conferma degli account utente senza la verifica dell'e-mail o del numero di telefono

Il trigger Lambda di pre-registrazione può essere utilizzato per confermare automaticamente gli account utente al momento della registrazione, senza richiedere un codice di conferma e verificare l'e-mail o il numero di telefono. Gli utenti che si confermano in questo modo possono effettuare l'accesso immediatamente senza la necessità di ricevere un codice.

Tramite questo trigger, puoi anche contrassegnare l'e-mail o il numero di telefono di un utente come verificati.

> **Note**
>
> Sebbene questo approccio sia utile per gli utenti che stanno appena cominciando, ti consigliamo di auto verificare almeno l'e-mail o il numero di telefono. In caso contrario, può accadere che l'utente non sia in grado di recuperare la password nel caso la dimentichi.

Se non richiedi che l'utente riceva e inserisca un codice di conferma durante la registrazione e non si esegue la verifica automatica di indirizzo e-mail e numero di telefono nel trigger Lambda di pre-registrazione, rischi di non avere un indirizzo e-mail o un numero di telefono verificato per quell'account utente. L'utente può verificare l'indirizzo e-mail o il numero di telefono in un secondo momento. Tuttavia, se l'utente dimentica la password e non dispone di un indirizzo e-mail o di un numero di telefono verificato, rimane bloccato fuori dall'account, in quanto il flusso di password dimenticata richiede un indirizzo e-mail o un numero di telefono verificato per inviare all'utente un codice di verifica.

Verifica in caso di modifica dell'e-mail o del numero di telefono da parte dell'utente

Quando un utente aggiorna il proprio indirizzo e-mail o numero di telefono nell'app, Amazon Cognito invia immediatamente un messaggio con un codice di verifica a tale utente se hai configurato il pool di utenti per la verifica automatica di tale attributo. L'utente deve quindi fornire all'app il codice incluso nel messaggio di verifica. L'app invia quindi il codice in una richiesta VerifyUserAttributeAPI per completare la verifica del nuovo valore dell'attributo.

Se il pool di utenti non richiede che questi ultimi verifichino un indirizzo e-mail o un numero di telefono aggiornato, Amazon Cognito cambia immediatamente il valore di un attributo `email` o `phone_number` aggiornato e contrassegna l'attributo come non verificato. L'utente non può accedere con un indirizzo e-mail o un numero di telefono non verificato. Deve infatti completare la verifica del valore aggiornato prima di poter utilizzare tale attributo come alias di accesso.

Se il pool di utenti richiede che questi ultimi verifichino un indirizzo e-mail o un numero di telefono aggiornato, Amazon Cognito mantieni l'attributo verificato e impostato sul valore originale finché l'utente non verifica il nuovo valore dell'attributo. Se l'attributo è un alias per l'accesso, l'utente può accedere con il valore originale dell'attributo fino a quando la verifica non cambia l'attributo nel nuovo valore. Per ulteriori informazioni sulla configurazione del pool di utenti per richiedere a questi ultimi di verificare gli attributi aggiornati, consulta Configurazione della verifica di e-mail o telefono.

Puoi utilizzare un trigger Lambda di messaggio personalizzato per personalizzare questo messaggio di verifica. Per ulteriori informazioni, consulta Trigger Lambda di messaggi personalizzati. Quando

l'indirizzo e-mail o il numero di telefono dell'utente non è verificato, l'app deve informare l'utente che deve verificare l'attributo. L'app deve inoltre rendere disponibile un pulsante o un link per la verifica del nuovo indirizzo e-mail o numero di telefono.

Processo di conferma e verifica degli account utente creati dagli amministratori o dagli sviluppatori

Gli account utente creati da un amministratore o da uno sviluppatore sono già in stato confermato, per cui agli utenti non viene richiesto di inserire un codice di conferma. Il messaggio di invito che il servizio Amazon Cognito invia a questi utenti include il nome utente e una password temporanea. L'utente deve modificare la password prima di effettuare l'accesso. Per ulteriori informazioni, consulta la Personalizzare e-mail ed SMS di Creazione di account utente come amministratore e il trigger di messaggio personalizzato in Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.

Processi di conferma e verifica per gli account utente importati

Gli account utente creati utilizzando la funzionalità di importazione degli utenti nella CLI o nell'API (vediImportazione di utenti nel bacino d'utenza da un file CSV) sono già nello stato confermato, quindi agli utenti non è richiesto di inserire un codice di conferma. AWS Management Console Non viene inviato alcun messaggio di invito. Tuttavia, gli account utente importati prevedono che gli utenti richiedano prima un codice chiamando l'API `ForgotPassword` e che in seguito creino una password utilizzando il codice inviato chiamando l'API `ConfirmForgotPassword` prima di effettuare l'accesso. Per ulteriori informazioni, consulta Necessità degli utenti importati di ripristinare le password.

L'e-mail o il numero di telefono devono essere contrassegnati come verificati quando l'account utente viene importato, in modo che non venga richiesta la verifica quando l'utente effettua l'accesso.

Invio di e-mail durante il test dell'App

Amazon Cognito invia agli utenti messaggi e-mail quando creano e gestiscono i propri account nell'app client per il pool di utenti. Se configuri il bacino d'utenza per richiedere la verifica e-mail, Amazon Cognito invia un'e-mail quando:

- Un utente effettua la registrazione.
- Un utente aggiorna il suo indirizzo e-mail.
- Un utente esegue un'operazione che chiama l'operazione API `ForgotPassword`.

- Crei un account utente come amministratore.

A seconda dell'azione che attiva l'e-mail, l'e-mail contiene un codice di verifica o una password temporanea. I tuoi utenti devono ricevere queste e-mail e comprendere il messaggio. In caso contrario, potrebbero non essere in grado di effettuare l'accesso e utilizzare la tua app.

Per essere che le e-mail vengano inviate senza problemi e che il messaggio sia corretto, prova le operazioni nella tua app attivando le consegne e-mail da Amazon Cognito. Ad esempio, utilizzando la pagina di registrazione nella tua app oppure utilizzando l'operazione API `SignUp`, puoi attivare un'e-mail effettuando la registrazione con un indirizzo e-mail di prova. Quando esegui il testing in questo modo, ricorda quanto segue:

> **Importante**
>
> Quando utilizzi un indirizzo e-mail per testare le operazioni che attivano le e-mail da Amazon Cognito, non utilizzare un indirizzo e-mail fittizio (ossia, senza una casella di posta associata). Utilizza un indirizzo e-mail reale che riceverà l'e-mail da Amazon Cognito senza creare un mancato recapito permanente.
> Un mancato recapito permanente si verifica quando Amazon Cognito non è in grado di consegnare le e-mail alla casella di posta del destinatario, cosa che accade sempre se la casella di posta è inesistente.
> Amazon Cognito limita il numero di e-mail che possono essere inviate da AWS account che subiscono ripetutamente hard bounce.

Quando provi le operazioni che attivano le e-mail, utilizza uno dei seguenti indirizzi e-mail per evitare gli hard bounce:

- Un indirizzo per un account e-mail di tua proprietà che utilizzi per il testing. Quando utilizzi il tuo indirizzo e-mail, riceverai un'e-mail inviata da Amazon Cognito. Con questa e-mail, puoi utilizzare il codice di verifica per testare l'esperienza di registrazione nella tua app. Se hai personalizzato il messaggio e-mail per il tuo bacino d'utenza, puoi controllare che le personalizzazioni siano corrette.
- L'indirizzo del simulatore di mailbox, success@simulator.amazonses.com. Se utilizzi l'indirizzo del simulatore, Amazon Cognito invia correttamente l'e-mail ma non sarai in grado di visualizzarla. Questa opzione è utile quando non è necessario utilizzare il codice di verifica né verificare il messaggio e-mail.

- L'indirizzo del simulatore di mailbox con un'etichetta arbitraria aggiuntiva, come success+user1@simulator.amazonses.com o success+user2@simulator.amazonses.com. Amazon Cognito invia correttamente e-mail a questi indirizzi ma non sarai in grado di visualizzarle. Questa opzione è utile quando si desidera testare il processo di registrazione aggiungendo più utenti di prova al bacino d'utenza e ogni utente di prova ha un indirizzo e-mail univoco.

Configurazione della verifica di e-mail o telefono

Puoi scegliere le impostazioni per la verifica tramite e-mail o telefono nella scheda Messaggistica. Per ulteriori informazioni sull'autenticazione a più fattori (Multi-Factor Authentication, MFA), consulta MFA con SMS.

Amazon Cognito utilizza Amazon SNS per inviare messaggi SMS. Se non hai mai inviato un messaggio SMS da Amazon Cognito o da altri in Servizio AWS precedenza, Amazon SNS potrebbe inserire il tuo account nella sandbox SMS. Ti consigliamo di inviare un messaggio di prova a un numero di telefono verificato prima di rimuovere il tuo account dalla sandbox e spostarlo alla produzione. Inoltre, se intendi inviare messaggi SMS a numeri di telefono di destinazione negli Stati Uniti, devi ottenere un ID mittente o di origine da Amazon Pinpoint. Per configurare il pool di utenti Amazon Cognito per i messaggi SMS, consulta Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.

Amazon Cognito è in grado di verificare automaticamente gli indirizzi e-mail e i numeri di telefono. Per effettuare questa verifica, Amazon Cognito invia un codice di verifica oppure un link di verifica. Per gli indirizzi e-mail, Amazon Cognito invia il codice o il link tramite un messaggio e-mail. Puoi scegliere un Tipo di verifica di Codice o Collegamento quando modifichi il modello Messaggi di verifica nella scheda Messaggistica della console Amazon Cognito. Per ulteriori informazioni, consulta Personalizzazione dei messaggi di verifica delle e-mail.

Per i numeri di telefono, Amazon Cognito invia il codice con un SMS.

Amazon Cognito deve verificare un numero di telefono o un indirizzo e-mail per confermare gli utenti e aiutarli a recuperare le password dimenticate. In alternativa, puoi confermare automaticamente gli utenti con il trigger Lambda di pre-registrazione o utilizzare AdminConfirmSignUpl'operazione API. Per ulteriori informazioni, consulta Registrazione e conferma degli account utente.

Il codice o link di verifica è valido 24 ore.

Se scegli di richiedere la verifica per un indirizzo e-mail o un numero di telefono, Amazon Cognito invia automaticamente il codice di verifica o il link quando un utente effettua la registrazione. Se il

bacino d'utenza ha un Trigger Lambda del mittente di SMS personalizzato o Trigger Lambda del mittente di e-mail personalizzato configurato, tale funzione viene invece richiamata.

> **Note**
>
> - L'utilizzo degli SMS per verificare i numeri di telefono viene addebitato separatamente da Amazon SNS. Per l'invio di messaggi e-mail, non viene addebitato alcun costo. Per le informazioni sui prezzi Amazon SNS, consulta la pagina Prezzi Worldwide SMS. Per la lista dei paesi in cui è disponibile la messaggistica SMS, consulta la pagina Supported regions and countries (regioni e paesi supportati).
> - Quando esegui il test delle operazioni nella tua app per la generazione di messaggi e-mail provenienti da Amazon Cognito, utilizza un indirizzo e-mail reale a cui Amazon Cognito può inviare messaggi senza incorrere in un mancato recapito permanente. Per ulteriori informazioni, consulta the section called “Invio di e-mail durante il test dell'App”.
> - Il flusso della reimpostazione della password richiede l'indirizzo e-mail o il numero di telefono dell'utente per verificarne l'identità.

> **Important**
>
> Se un utente si registra sia con un numero di telefono sia con un indirizzo e-mail e le impostazioni del pool di utenti richiedono la verifica di entrambi gli attributi, Amazon Cognito invia un codice di verifica tramite SMS al numero di cellulare. Amazon Cognito non ha ancora verificato l'indirizzo e-mail, quindi l'app deve chiamare GetUserper vedere se un indirizzo e-mail è in attesa di verifica. Se richiede una verifica, l'app deve chiamare GetUserAttributeVerificationCodeper avviare il flusso di verifica dell'e-mail. Quindi deve inviare il codice di verifica VerifyUserAttributechiamando.

Puoi modificare la quota di spesa per i messaggi SMS per uno Account AWS e per singoli messaggi. I limiti si applicano solo al costo di invio di messaggi SMS. Per ulteriori informazioni, consulta Quali sono i limiti di spesa a livello di account e a livello di messaggio e come funzionano? in Domande frequenti su Amazon SNS.

Amazon Cognito invia messaggi SMS utilizzando le risorse Amazon SNS nel luogo in cui è stato creato Regione AWS il pool di utenti o in una regione alternativa legacy di Amazon SNS, come indicato nella tabella seguente. L'eccezione è rappresentata dai bacini d'utenza Amazon Cognito

nella regione Asia Pacifico (Seoul). Questi bacini d'utenza utilizzano la tua configurazione Amazon SNS nella regione Asia Pacifico (Tokyo). Per ulteriori informazioni, consulta Scegli l'opzione Regione AWS per i messaggi SMS di Amazon SNS.

Regione di Amazon Cognito	Regione alternativa Amazon SNS legacy
Stati Uniti orientali (Ohio)	Stati Uniti orientali (Virginia settentrionale)
Asia Pacifico (Mumbai)	Asia Pacifico (Singapore)
Asia Pacifico (Seoul)	Asia Pacifico (Tokyo)
Canada (Centrale)	Stati Uniti orientali (Virginia settentrionale)
Europa (Francoforte)	Europa (Irlanda)
Europa (Londra)	Europa (Irlanda)

Esempio: se il tuo bacino d'utenza Amazon Cognito si trova in Asia Pacifico (Mumbai) e hai aumentato il tuo limite di spesa in ap-southeast-1, forse non ti conviene richiedere un aumento separato in ap-south-1. Piuttosto, puoi utilizzare le tue risorse Amazon SNS in Asia Pacifico (Singapore).

Verifica degli aggiornamenti a indirizzi e-mail e numeri di telefono

Un attributo di indirizzo e-mail o un numero di telefono può diventare attivo e non verificato subito dopo che l'utente ha modificato il relativo valore. Amazon Cognito può anche richiedere che l'utente verifichi il nuovo valore prima che Amazon Cognito aggiorni l'attributo. Quando è necessario che gli utenti verifichino innanzitutto il nuovo valore, possono utilizzare il valore originale per l'accesso e per ricevere messaggi fino a quando non verificheranno il nuovo valore.

Quando gli utenti possono utilizzare il proprio indirizzo e-mail o numero di telefono come alias di accesso nel pool di utenti, il nome di accesso per un attributo aggiornato dipende dall'eventuale richiesta di verifica degli attributi aggiornati. Quando si richiede che gli utenti verifichino un attributo aggiornato, un utente può accedere con il valore originale dell'attributo fino a quando non verifica il nuovo valore. Quando non si richiede che gli utenti verifichino un attributo aggiornato, un utente non può accedere o ricevere messaggi con il valore nuovo o originale fino a quando non verifica il nuovo valore.

Ad esempio, il pool di utenti consente l'accesso con un alias di indirizzo e-mail e richiede che gli utenti verifichino il proprio indirizzo e-mail al momento dell'aggiornamento. Sue, che ha effettuato l'accesso come `sue@example.com`, vuole cambiare il suo indirizzo email in `sue2@example.com` ma per errore immette **s**`sue2@example.com`. Sue non riceve l'e-mail di verifica, quindi non può verificare `ssue2@example.com`. Sue accede come `sue@example.com` e invia nuovamente il modulo nell'app per aggiornare il suo indirizzo e-mail in `sue2@example.com`. Riceve il messaggio e-mail, fornisce il codice di verifica all'app e inizia ad accedere come `sue2@example.com`.

Quando un utente aggiorna un attributo e il pool di utenti verifica i nuovi valori degli attributi

- Possono accedere con il valore dell'attributo originale prima di aver confermato il codice per verificare il nuovo valore.
- Possono accedere solo con il valore dell'attributo originale dopo aver confermato il codice per verificare il nuovo valore.
- Se `phone_number_verified` imposti `email_verified` o `true` abiliti una richiesta AdminUpdateUserAttributesAPI, possono accedere prima di aver confermato il codice che Amazon Cognito gli ha inviato.

Quando un utente aggiorna un attributo e il pool di utenti non verifica i nuovi valori degli attributi

- Non possono accedere né ricevere messaggi utilizzando il valore dell'attributo originale.
- Non possono accedere né ricevere messaggi diversi da un codice di conferma utilizzando il nuovo valore dell'attributo prima di aver confermato il codice per verificare il nuovo valore.
- Se `phone_number_verified` imposti `email_verified` o `true` abiliti una richiesta AdminUpdateUserAttributesAPI, possono accedere prima di aver confermato il codice che Amazon Cognito gli ha inviato.

Richiedere la verifica dell'attributo quando gli utenti aggiornano l'indirizzo e-mail o numero di telefono

1. Accedi alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Nella scheda Esperienza di registrazione scegli Modifica in Attribute verification and user account confirmation (Verifica attributi e conferma account utente).
4. Scegli Keep original attribute value active when an update is pending (Mantieni attivo il valore dell'attributo originale quando un aggiornamento è in sospeso).

5. In Active attribute values when an update is pending (Valori di attributo attivi quando un aggiornamento è in sospeso), scegli gli attributi che gli utenti devono di verificare prima che Amazon Cognito aggiorni il valore.
6. Seleziona Salvataggio delle modifiche.

Per richiedere la verifica dell'aggiornamento degli attributi con l'API Amazon Cognito, puoi impostare il `AttributesRequireVerificationBeforeUpdate` parametro in una UpdateUserPoolrichiesta.

Autorizzazione di Amazon Cognito a inviare SMS per tuo conto

Per poter inviare SMS agli utenti per tuo conto, Amazon Cognito ha bisogno della tua autorizzazione. Per concedere tale autorizzazione, puoi creare un ruolo AWS Identity and Access Management (IAM). Nella scheda Messaggistica della console Amazon Cognito, in SMS, scegli Modifica per impostare un ruolo.

Configurazione dei messaggi di verifica di SMS e-mail e dei messaggi di invito degli utenti

Amazon Cognito ti consente di personalizzare i messaggi di verifica SMS ed e-mail, nonché i messaggi di invito degli utenti, per migliorare la sicurezza e l'esperienza utente della tua applicazione. Con Amazon Cognito, puoi scegliere tra verifiche basate sul codice o sui link con un clic per soddisfare le esigenze della tua applicazione. Questo argomento spiega come personalizzare l'autenticazione a più fattori (MFA) e le comunicazioni di verifica nella console Amazon Cognito.

Nella scheda Messaggistica, in Modelli di messaggio, puoi personalizzare:

- Gli SMS di autenticazione a più fattori (MFA)
- I messaggi di verifica degli SMS e delle e-mail
- Il tipo di verifica dell'e-mail, ovvero codice o link
- I messaggi di invito degli utenti
- Gli indirizzi e-mail FROM e REPLY-TO delle e-mail del bacino d'utenza

> **Note**
>
> I modelli dei messaggi di verifica degli SMS e delle e-mail vengono visualizzati solo se hai scelto di richiedere la verifica del numero di telefono e dell'e-mail nella scheda Verifications

(Verifiche). Analogamente, il modello di messaggio MFA SMS viene visualizzato solo se le impostazioni MFA sono obbligatorie o facoltative.

Argomenti

- Modelli dei messaggi
- Personalizzazione del messaggio SMS
- Personalizzazione dei messaggi di verifica delle e-mail
- Personalizzazione dei messaggi di invito degli utenti
- Personalizzazione dell'indirizzo e-mail
- Autorizzazione di Amazon Cognito a inviare e-mail di Amazon SES per tuo conto (da un indirizzo e-mail FROM personalizzato)

Modelli dei messaggi

I modelli dei messaggi ti consentono di inserire un campo nel messaggio utilizzando un segnaposto che viene sostituito con il valore corrispondente.

Segnaposti dei modelli

Descrizione	Token
Codice di verifica	`{####}`
Password temporanea	`{####}`
Nome utente	`{username}`

> **Note**
>
> Non è possibile utilizzare il segnaposto `{username}` nei messaggi e-mail di verifica. Puoi utilizzare il `{username}` segnaposto nei messaggi e-mail di invito generati con l'operazione. AdminCreateUser Questi messaggi e-mail di invito utilizzano due segnaposto: il nome utente, come `{username}` e la password temporanea come `{####}`.

Puoi utilizzare i segnaposti dei modelli di sicurezza avanzata per:

- Includere dettagli specifici su un evento come l'indirizzo IP, la città, il paese, l'ora di accesso e il nome del dispositivo. Le funzionalità di sicurezza avanzate di Amazon Cognito possono analizzare questi dettagli.
- Verificare se un link con un clic è valido.
- Usare l'ID dell'evento, il token di feedback e il nome utente per costruire il tuo link con un solo clic.

> **Note**
>
> Per generare link con un clic e utilizzare i segnaposti `{one-click-link-valid}` e `{one-click-link-invalid}` nei modelli di e-mail di sicurezza avanzati, devi disporre già di un dominio configurato per il pool di utenti.

Segnaposti dei modelli di sicurezza avanzata

Descrizione	Token
Indirizzo IP	`{ip-address}`
City	`{city}`
Paese	`{country}`
Orario di accesso	`{login-time}`
Nome dispositivo	`{device-name}`
Il link con un clic è valido	`{one-click-link-valid}`
Il link con un clic non è valido	`{one-click-link-invalid}`
ID evento	`{event-id}`
Token del feedback	`{feedback-token}`

Personalizzazione del messaggio SMS

> **Note**
>
> Nella nuova esperienza della console di Amazon Cognito, puoi personalizzare i messaggi SMS.

Puoi personalizzare il messaggio SMS per l'autenticazione a più fattori (MFA) nella scheda Messaggistica nell'intestazione Modelli di messaggio.

> **Important**
>
> Il tuo messaggio personalizzato deve contenere il segnaposto `{####}`. Questo segnaposto verrà sostituito con il codice di autenticazione al momento dell'invio del messaggio.

Amazon Cognito impone una lunghezza massima per i messaggi SMS, incluso il codice di autenticazione, pari a 140 caratteri UTF-8.

Personalizzazione dei messaggi di verifica degli SMS

Puoi personalizzare il messaggio SMS per la verifica del numero di telefono modificando il modello all'intestazione Do you want to customize your SMS verification messages? (Vuoi personalizzare i messaggi di verifica?).

> **Important**
>
> Il tuo messaggio personalizzato deve contenere il segnaposto `{####}`. Questo segnaposto verrà sostituito con il codice di verifica al momento dell'invio del messaggio.

La lunghezza massima per il messaggio è pari a 140 caratteri UTF-8, incluso il codice di verifica.

Personalizzazione dei messaggi di verifica delle e-mail

Per verificare l'indirizzo e-mail di un utente nel bacino d'utenza con Amazon Cognito, puoi inviare all'utente un messaggio e-mail con un link selezionabile, oppure un codice da inserire.

Per personalizzare l'oggetto e il contenuto del messaggio e-mail per i messaggi di verifica dell'indirizzo e-mail, modifica il modello Messaggi di verifica nella scheda Messaggistica del pool

di utenti. Puoi scegliere un Tipo di verifica di Codice o Collegamento quando modifichi il modello Messaggi di verifica.

Quando scegli Codice come tipo di verifica, il messaggio personalizzato deve contenere il segnaposto `{####}`. Quando invii il messaggio, il segnaposto viene sostituito con il codice di verifica.

Quando scegli Collegamento come tipo di verifica, il messaggio personalizzato deve includere un segnaposto nel formato `{##Verify Your Email##}`. Puoi modificare la stringa di testo tra i caratteri segnaposto, ad esempio `{##Click here##}`. Il segnaposto viene sostituito da un link di verifica con il testo Verifica il tuo indirizzo e-mail.

Il link per un messaggio di verifica e-mail indirizza l'utente a un URL come nell'esempio seguente.

```
https://<your user pool domain>/confirmUser/?
client_id=abcdefg12345678&user_name=emailtest&confirmation_code=123456
```

La lunghezza massima per il messaggio è pari a 20.000 caratteri UTF-8, incluso il codice di verifica (se presente). È possibile utilizzare i tag HTML in questo messaggio per formattare i contenuti.

Personalizzazione dei messaggi di invito degli utenti

Puoi personalizzare il messaggio di invito degli utenti che Amazon Cognito invia ai nuovi utenti tramite messaggio SMS o e-mail modificando il modello Messaggi di invito nella scheda Messaggistica.

> **Important**
>
> Il tuo messaggio personalizzato deve contenere i segnaposti `{username}` e `{####}`. Quando Amazon Cognito invia il messaggio di invito, sostituisce questi segnaposti con il nome utente e la password dell'utente.

La lunghezza massima di un messaggio SMS è pari a 140 caratteri UTF-8, incluso il codice di verifica. La lunghezza massima di un messaggio e-mail è pari a 20.000 caratteri UTF-8, incluso il codice di verifica. È possibile utilizzare tag HTML nei messaggi e-mail per formattare il contenuto.

Personalizzazione dell'indirizzo e-mail

Per impostazione predefinita, i messaggi e-mail che Amazon Cognito invia agli utenti nel bacino d'utenza provengono dall'indirizzo no-reply@verificationemail.com. Puoi specificare indirizzi e-mail personalizzati FROM e REPLY-TO da usare al posto di no-reply@verificationemail.com.

Per personalizzare gli indirizzi e-mail del mittente (FROM) e del destinatario (REPLY-TO)

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Messaggistica. Alla voce E-mail, scegli Modifica.
4. Scegli una Regione SES.
5. Scegli un indirizzo e-mail FROM dall'elenco degli indirizzi e-mail verificati con Amazon SES nella Regione SES selezionata. Per utilizzare un indirizzo e-mail da un dominio verificato, configura le impostazioni e-mail in AWS Command Line Interface o nelle API AWS. Per ulteriori informazioni, consulta Verifica indirizzi e-mail e domini in Amazon SES nella Guida per gli sviluppatori di Amazon SES.
6. Scegli un Set di configurazione dall'elenco dei set di configurazione nella Regione SES selezionata.
7. Inserisci un nome mittente FROM cordiale per i tuoi messaggi e-mail nel formato `John Stiles <johnstiles@example.com>`.
8. Per personalizzare l'indirizzo e-mail del destinatario, inserisci un indirizzo e-mail valido nel campo Indirizzo e-mail destinatario.

Autorizzazione di Amazon Cognito a inviare e-mail di Amazon SES per tuo conto (da un indirizzo e-mail FROM personalizzato)

Puoi configurare Amazon Cognito per inviare e-mail da un indirizzo e-mail FROM personalizzato anziché dal relativo indirizzo predefinito. Per utilizzare un indirizzo personalizzato, devi concedere ad Amazon Cognito l'autorizzazione ad inviare un messaggio e-mail da un'identità verificata di Amazon SES. Nella maggior parte dei casi, per concedere tale autorizzazione crea una policy di autorizzazione all'invio. Per ulteriori informazioni, consulta Utilizzo dell'autorizzazione di invio con Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Quando configuri un bacino d'utenza per utilizzare Amazon SES per i messaggi e-mail, Amazon Cognito crea il ruolo `AWSServiceRoleForAmazonCognitoIdpEmailService` nel tuo account per concedere l'accesso ad Amazon SES. Non è necessario alcuna policy di autorizzazione per l'invio quando viene utilizzato il `AWSServiceRoleForAmazonCognitoIdpEmailService` ruolo collegato ai servizi. È necessario aggiungere una policy di autorizzazione per l'invio solo quando si utilizzano entrambe le funzionalità e-mail predefinite nel bacino d'utenza e un'identità Amazon SES verificata come indirizzo FROM.

Per ulteriori informazioni sul ruolo collegato al servizio creato da Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

L'esempio seguente di policy di autorizzazione di invio garantisce ad Amazon Cognito la possibilità limitata di utilizzare un'identità verificata di Amazon SES. Amazon Cognito può inviare e-mail solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `aws:SourceAccount`. Per altri esempi, consulta Esempi di policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

> **Note**
>
> In questo esempio, il valore "Sid" è una stringa arbitraria che identifica in modo univoco l'istruzione. Per ulteriori informazioni sulla sintassi della policy, consulta Policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

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

```
        }
    ]
}
```

La console Amazon Cognito aggiunge una policy del genere quando viene selezionata un'identità Amazon SES dal menu a discesa. Se utilizzi la CLI o l'API per configurare il bacino d'utenza, è necessario allegare una policy strutturata in modo simile all'esempio precedente alla tua identità di Amazon SES.

Creazione di account utente come amministratore

Dopo aver creato il bacino d'utenza, puoi creare gli utenti utilizzando la AWS Management Console, AWS Command Line Interface oppure l'API di Amazon Cognito. Puoi creare il profilo di un nuovo utente in un bacino d'utenza e inviargli un messaggio di benvenuto con le istruzioni di registrazione tramite SMS o e-mail.

Gli sviluppatori e gli amministratori possono eseguire le seguenti attività:

- Creare un nuovo profilo utente utilizzando la AWS Management Console o chiamando l'API `AdminCreateUser`.
- Impostazione dei valori degli attributi utente.
- Creazione di attributi personalizzati.
- Imposta il valore degli attributi personalizzati non modificabili nelle richieste API `AdminCreateUser`. Questa funzionalità non è disponibile nella console Amazon Cognito.
- Specifica la password temporanea o consenti ad Amazon Cognito di generarne una automaticamente.
- Specificare se i numeri di telefono e gli indirizzi e-mail forniti sono stati contrassegnati come verificati per i nuovi utenti.
- Specificare i messaggi di invito personalizzati con e-mail ed SMS per i nuovi utenti tramite la AWS Management Console o un trigger Lambda di messaggio personalizzato. Per ulteriori informazioni, consulta Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda.
- Specificare se i messaggi di invito vengono inviati tramite SMS, e-mail, o entrambi.
- Inviare nuovamente il messaggio di benvenuto a un utente esistente chiamando l'API `AdminCreateUser`, specificando `RESEND` per il parametro `MessageAction`.

> **Note**
>
> Questa azione attualmente non può essere eseguita utilizzando la AWS Management Console.

- Sopprimere l'invio del messaggio di invito al momento della creazione dell'utente.
- Specificare il limite di tempo della scadenza dell'account utente (fino a 90 giorni).
- Consentire agli utenti di registrarsi o richiedere che i nuovi utenti vengano aggiunti solo dall'amministratore.

Flussi di autenticazione degli utenti creati dagli amministratori o dagli sviluppatori

Il flusso di autenticazione di questi utenti prevede una fase ulteriore, vale a dire inviare la nuova password e fornire i valori mancanti degli attributi obbligatori. Dette fasi sono di seguito descritte; i punti 5, 6 e 7 sono specifici per questi utenti.

1. L'utente inizia il primo accesso inviando il nome utente e la password.
2. L'SDK chiama `InitiateAuth(Username, USER_SRP_AUTH)`.
3. Amazon Cognito restituisce la sfida `PASSWORD_VERIFIER` con il blocco Salt & Secret.
4. L'SDK esegue i calcoli SRP e chiama `RespondToAuthChallenge(Username, <SRP variables>, PASSWORD_VERIFIER)`.
5. Amazon Cognito restituisce la sfida `NEW_PASSWORD_REQUIRED`. Il corpo di questa sfida include gli attributi correnti dell'utente e tutti gli attributi richiesti nel pool di utenti che attualmente non hanno un valore nel profilo dell'utente. Per ulteriori informazioni, consulta RespondToAuthChallenge.
6. All'utente viene richiesto di immettere una nuova password e i valori mancanti degli attributi obbligatori.
7. L'SDK chiama `RespondToAuthChallenge(Username, <New password>, <User attributes>)`.
8. Se l'utente necessita di un secondo fattore per MFA, Amazon Cognito restituisce la sfida SMS_MFA e il codice viene inviato.
9. Dopo che l'utente ha modificato correttamente la propria password e, facoltativamente, fornito i valori attribuiti o completato l'MFA, è in grado di effettuare l'accesso e i token vengono rilasciati.

Quando l'utente ha soddisfatto tutte le sfide, il servizio Amazon Cognito lo contrassegna come confermato ed emette i token di ID, accesso e aggiornamento per l'utente. Per ulteriori informazioni, consulta Utilizzo di token con bacini d'utenza.

Creazione di un nuovo utente nella AWS Management Console

Puoi impostare i requisiti della password utente, configurare i messaggi di invito e verifica inviati agli utenti e aggiungere nuovi utenti con la console Amazon Cognito.

Impostare una policy per le password e abilitare l'auto-registrazione

È possibile configurare le impostazioni per password dalla complessità minima e decidere se gli utenti possono registrarsi utilizzando le API pubbliche nel bacino d'utenza.

Configurare una policy per le password

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Sign-in experience (esperienza di accesso) e trova le Password policy (policy per le password). Scegli Modifica.
4. Impostare la Password policy mode (modalità di policy per le password) su Custom (personalizzata).
5. Scegli una lunghezza minima della password. Per i limiti al requisito di lunghezza della password, consulta User pools resource quotas (Quote di risorse del pool di utenti).
6. Scegli un requisito di Password complexity (complessità delle password).
7. Scegli la durata della validità della password impostata dagli amministratori.
8. Scegli Save changes (salva modifiche).

Consenti l'iscrizione self-service

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Sign-up experience (esperienza di registrazione) e trova Self-service sign-up (registrazione self-service). Seleziona Edit (Modifica).
4. Decidi se scegliere l'opzione Abilita l'auto-registrazione. L'auto-registrazione viene solitamente utilizzata con client di app pubblici che devono registrare nuovi utenti nel tuo bacino d'utenza

senza distribuire un segreto client o credenziali API AWS Identity and Access Management (IAM).

> **Disattivare dell'auto-registrazione**
>
> Se non abiliti l'auto-registrazione, i nuovi utenti devono essere creati mediante operazioni API amministrative utilizzando le credenziali API IAM o tramite accesso con i provider federati.

5. Seleziona Salva modifiche.

Personalizzare e-mail ed SMS

Personalizzare i messaggi utente

Puoi personalizzare i messaggi inviati da Amazon Cognito ai tuoi utenti che ricevono un invito ad accedere, si registrano per un account utente o a cui viene richiesta l'autenticazione a più fattori (MFA) all'accesso.

> **Note**
>
> Viene inviato un Invitation message (messaggio di invito) quando crei un utente nel bacino d'utenza e lo inviti a effettuare l'accesso. Amazon Cognito invia le informazioni di accesso iniziali all'indirizzo e-mail o al numero di telefono dell'utente.
> Quando un utente effettua la registrazione per un account utente nel bacino d'utenza, viene inviato un messaggio di verifica. Amazon Cognito invia un codice all'utente. Quando l'utente fornisce il codice ad Amazon Cognito, verifica le informazioni di contatto e conferma il proprio account per l'accesso. I codici di verifica sono validi 24 ore.
> Quando si abilita la MFA via SMS nel bacino d'utenza e un utente che ha configurato la MFA via SMS accede e gli viene richiesto di effettuare la MFA, viene inviato un Messaggio MFA.

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Messaging (messaggistica) e ricerca Message templates (modelli di messaggi). Seleziona Verification messages (messaggi di verifica), Invitation messages (messaggi di invito), oppure Messaggi MFA e scegli Edit (modifica).

4. Personalizza i messaggi per il tipo di messaggio scelto.

> **Note**
>
> Tutte le variabili nei modelli di messaggio devono essere incluse quando si personalizza il messaggio. Se la variabile, ad esempio {####}, non è inclusa, l'utente avrà informazioni insufficienti per completare l'azione del messaggio.
> Per ulteriori informazioni, vedi Message templates (modelli di messaggi).

5. a. Verification messages (Messaggi di verifica)

    i. Scegli un Verification type (tipo di verifica) per le e-mail. Un Code (codice) di verifica invia un codice numerico che l'utente dovrà inserire. Un Link di verifica invia un link su cui l'utente può fare clic per verificare le informazioni di contatto. Il testo nella variabile per un messaggio Link viene visualizzato come testo del collegamento ipertestuale. Ad esempio, viene visualizzato un modello di messaggio che utilizza la variabile {##Click here##} Click here (clicca qui) nel testo dell'e-mail.

    ii. Specificare un Email subject (oggetto dell'e-mail) per i messaggi via e-mail.

    iii. Inserisci un modello di e-mail personalizzato per i messaggi via e-mail. È possibile personalizzare questo modello in formato HTML.

    iv. Inserisci un modello personalizzato di SMS per i messaggi SMS.

    v. Scegli Save changes (salva modifiche).

    b. Invitation messages (Messaggi di invito)

    i. Specificare un Email subject (oggetto dell'e-mail) per i messaggi via e-mail.

    ii. Inserisci un modello di e-mail personalizzato per i messaggi via e-mail. È possibile personalizzare questo modello in formato HTML.

    iii. Inserisci un modello personalizzato di SMS per i messaggi SMS.

    iv. Scegli Save changes (salva modifiche).

    c. MFA messages (Messaggi MFA)

    i. Inserisci un modello personalizzato di SMS per i messaggi SMS.

    ii. Scegli Save changes (salva modifiche).

Creazione di un utente

Creazione di un utente

Puoi creare nuovi utenti per il tuo bacino d'utenza dalla console Amazon Cognito. In genere, gli utenti possono accedere dopo aver impostato una password. Per effettuare l'accesso con un indirizzo e-mail l'utente deve verificare l'attributo `email`. Per accedere con un numero di telefono, l'utente deve verificare l'attributo `phone_number`. È inoltre possibile confermare gli account come amministratore utilizzando AWS CLI o l'API oppure creando profili utente con un provider di identità federato. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di Amazon Cognito.

1. Passa alla console Amazon Cognito e scegli bacini d'utenza.
2. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
3. Scegli la scheda Users (Utenti), quindi seleziona Create a user (Crea un utente).
4. Esamina i User pool sign-in and security requirements (Requisiti di accesso e sicurezza del bacino d'utenza) per informazioni sui requisiti delle password, sui metodi di recupero dell'account disponibili e sugli attributi alias per il bacino d'utenza.
5. Scegli come desideri inviare l'Invitation message (Messaggio d'invito). Puoi scegliere messaggio SMS, messaggio e-mail o entrambi.

   > **Note**
   >
   > Prima di inviare i messaggi di invio, configura un mittente e una Regione AWS con Amazon Simple Notification Service e Amazon Simple Email Service nella scheda Messaging (Messaggistica) del bacino d'utenza. Messaggi e velocità dati del destinatario. Amazon SES fattura separatamente i messaggi e-mail e Amazon SNS fattura separatamente i messaggi SMS.

6. Scegli uno Username (nome utente) per il nuovo utente.
7. Scegli se selezionare Create a password (Crea una password) o se vuoi che Amazon Cognito esegua l'operazione Generate a password (Genera una password) per l'utente. Qualsiasi password temporanea deve rispettare i criteri delle password del bacino d'utenza.
8. Seleziona Crea.
9. Seleziona la scheda Users (Utenti) e scegli il valore User name (Nome utente) per l'utente. Aggiungi e modifica User attributes (Attributi utente) e Group memberships (Appartenenze a gruppi). Esamina User event history (Cronologia eventi dell'utente).

Aggiunta di gruppi a un bacino d'utenza

Il supporto per i gruppi nei bacini d'utenza di Amazon Cognito ti permette di creare e gestire gruppi e di aggiungere e rimuovere gli utenti dai gruppi. Utilizza i gruppi per creare raccolte di utenti e gestire le loro autorizzazioni o rappresentare diversi tipi di utenti. Puoi assegnare un ruolo AWS Identity and Access Management (IAM) a un gruppo per definire le autorizzazioni per i membri di un gruppo.

Puoi utilizzare i gruppi per creare una raccolta di utenti in un bacino d'utenza, operazione che spesso viene svolta per impostare le autorizzazioni di tali utenti. Ad esempio, è possibile creare gruppi separati per utenti che siano lettori, collaboratori e revisori del tuo sito Web e della tua app. Utilizzando il ruolo IAM associato a un gruppo, puoi impostare anche autorizzazioni diverse per questi diversi gruppi, in modo che solo i collaboratori possano inserire dei contenuti in Amazon S3 e solo gli editor possano pubblicare dei contenuti tramite un'API in Amazon API Gateway.

È possibile creare e gestire gruppi in un pool di utenti da AWS Management Console, le API e la CLI. In qualità di sviluppatore (utilizzando AWS le credenziali), puoi creare, leggere, aggiornare, eliminare ed elencare i gruppi per un pool di utenti. Puoi inoltre aggiungere e rimuovere gli utenti dai gruppi.

Non sono previsti costi aggiuntivi per l'utilizzo di gruppi all'interno di un bacino d'utenza. Per ulteriori informazioni, consulta Prezzi di Amazon Cognito.

Assegnazione di ruoli IAM ai gruppi

È possibile utilizzare i gruppi per controllare le autorizzazioni per le risorse utilizzando un ruolo IAM. I ruoli IAM includono policy di attendibilità e policy di autorizzazione. La policy di attendibilità del ruolo specifica chi può utilizzare il ruolo. Le policy di autorizzazioni specificano le operazioni e le risorse a cui i membri del gruppo possono accedere. Quando crei un ruolo IAM, imposta la policy di attendibilità del ruolo per consentire agli utenti del gruppo di assumere il ruolo. Nelle policy delle autorizzazioni del ruolo specificare le autorizzazioni che si desidera concedere al gruppo.

Quando crei un gruppo in Amazon Cognito, specifichi un ruolo IAM fornendo l'ARNdel ruolo. Quando i membri del gruppo accedono utilizzando Amazon Cognito, possono ricevere credenziali temporanee dai pool di identità. Le autorizzazioni sono determinate dal ruolo IAM associato.

I singoli utenti possono essere in più gruppi. In qualità di sviluppatore, hai a disposizione le seguenti opzioni per scegliere automaticamente il ruolo IAM quando un utente si trova in più gruppi:

- Puoi assegnare i valori di priorità per ciascun gruppo. Verrà scelto il gruppo con la priorità migliore (più bassa) e sarà applicato il relativo ruolo IAM associato.

- L'app può anche scegliere tra i ruoli disponibili quando richiede AWS le credenziali per un utente tramite un pool di identità, specificando un ruolo ARN nel parametro. GetCredentialsForIdentity`CustomRoleARN` Il ruolo IAM specificato deve corrispondere a un ruolo disponibile per l'utente.

Assegnazione dei valori di priorità ai gruppi

Un utente può appartenere a più di un gruppo. Nei token di accesso e ID dell'utente, l'attestazione `cognito:groups` contiene l'elenco di tutti i gruppi a cui un utente appartiene. L'attestazione `cognito:roles` contiene l'elenco dei ruoli corrispondenti ai gruppi.

Poiché un utente può appartenere a più di un gruppo, puoi assegnare una priorità a ciascun gruppo. Questo è un numero non negativo che specifica la priorità di questo gruppo rispetto agli altri gruppi ai quali un utente appartiene nel bacino d'utenza. Zero è il valore di priorità massimo. I gruppi con valori di priorità più bassi prevalgono sui gruppi con valori di priorità più alti o nulli. Se un utente appartiene a due o più gruppi, verrà scelto il gruppo il cui ruolo IAM abbia il valore di priorità più basso applicato all'attestazione `cognito:preferred_role` nel token ID dell'utente.

Due gruppi possono avere lo stesso valore di priorità. In questo caso nessun gruppo prevale sull'altro. Se due gruppi con lo stesso valore di priorità hanno lo stesso ruolo ARN, tale ruolo viene utilizzato nell'attestazione `cognito:preferred_role` nei token ID per gli utenti di ciascun gruppo. Se i due gruppi hanno diversi ruoli ARN, l'attestazione `cognito:preferred_role` non è impostata nei token ID degli utenti.

Utilizzo di gruppi per controllare l'autorizzazione con Amazon API Gateway

Puoi utilizzare i gruppi di un bacino d'utenza per controllare l'autorizzazione con Amazon API Gateway. I gruppi di cui un utente è membro sono inclusi sia nel token ID che nel token di accesso da un bacino d'utenza nell'attestazione `cognito:groups`. Puoi inviare ID o token di accesso con richieste ad Amazon API Gateway e utilizzare un autorizzazione del bacino d'utenza Amazon Cognito per un'API REST. Per ulteriori informazioni, consulta la sezione Control access to a REST API using Amazon Cognito user pools as authorizer (Controllo degli accessi a un'API REST utilizzando pool di utenti di Amazon Cognito come autorizzazione) nella Guida per sviluppatori di API Gateway.

È inoltre possibile autorizzare l'accesso a un'API HTTP di Amazon API Gateway con un'autorizzazione JWT personalizzata. Per ulteriori informazioni consulta la sezione Controlling access to HTTP APIs with JWT authorizers (Controllo dell'accesso alle API HTTP con le autorizzazioni JWT) nella Guida per gli sviluppatori dell'API Gateway.

Limitazioni per i gruppi

I gruppi di utenti sono soggetti alle seguenti limitazioni:

- Il numero di gruppi che puoi creare è limitato dalle quote del servizio Amazon Cognito.
- I gruppi non possono essere annidati.
- Non puoi cercare gli utenti in un gruppo.
- Non puoi cercare i gruppi per nome, ma puoi farne un elenco.

Creazione di un nuovo gruppo nella AWS Management Console

Per creare un nuovo gruppo, utilizza la procedura seguente.

Per creare un nuovo gruppo

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Groups (gruppi) quindi seleziona Create a group (crea gruppo).
5. Nella pagina Create a group (Crea gruppo), sotto Group name (nome gruppo) inserisci un nome per il gruppo.
6. Facoltativamente, è possibile fornire ulteriori informazioni su questo gruppo utilizzando uno dei seguenti campi:

   - Description (Descrizione) - Inserisci i dettagli sull'utilizzo futuro di questo nuovo gruppo.
   - Precedence (Priorità) - Amazon Cognito valuta e applica tutte le autorizzazioni di gruppo per un determinato utente in base ai gruppi a cui appartengono hanno un valore di precedenza inferiore. Verrà scelto il gruppo con la priorità più bassa e sarà applicato il relativo ruolo IAM associato. Per ulteriori informazioni, consulta Assegnazione dei valori di priorità ai gruppi.
   - Ruolo IAM - È possibile assegnare un ruolo IAM al gruppo quando è necessario controllare le autorizzazioni per le risorse. In caso di integrazione di un bacino d'utenza con un pool di identità, l'impostazione IAM role (ruolo IAM) determina il ruolo da assegnare nel token ID dell'utente se il pool di identità è configurato per la scelta del ruolo dal token. Per ulteriori informazioni, consulta Assegnazione di ruoli IAM ai gruppi.
   - Add users to this group (Aggiungi utenti a questo gruppo) - Aggiungi utenti esistenti come membri di questo gruppo dopo la creazione.

7. Scegli Create (Crea) per confermare.

Gestione e ricerca degli account utente

Una volta creato il bacino d'utenza, potrai visualizzare e gestire gli utenti tramite la AWS Management Console, la AWS Command Line Interface oppure l'API di Amazon Cognito. Questo argomento descrive in che modo puoi visualizzare e cercare gli utenti tramite la AWS Management Console.

Visualizzazione degli attributi utente

Utilizza la procedura seguente per visualizzare gli attributi utente nella console Amazon Cognito.

Per visualizzare gli attributi utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Users (utenti), quindi seleziona un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, alla voce User attributes (attributi utente), è possibile visualizzare quali attributi siano associati all'utente.

Reimpostare una password utente

Utilizza la procedura seguente per reimpostare la password utente nella console Amazon Cognito.

Per reimpostare una password utente

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Users (utenti), quindi scegli un utente nell'elenco.
5. Nella pagina dei dettagli dell'utente, scegli Actions (Operazioni),Reset password (reimposta password).
6. Nella finestra di dialogo Reset password (reimposta password), controlla le informazioni e, quando sei pronto, scegli Reimposta.

L'operazione genera un codice di conferma che viene inviato all'utente e ne disabilita la password corrente cambiando lo stato dell'utente in `RESET_REQUIRED`. Il codice Reset password (Reimposta password) è valido per 1 ora.

Ricerca degli attributi utente

Se hai già creato un bacino d'utenza, puoi effettuare la ricerca dal pannello Users (Utenti) nella AWS Management Console. Puoi utilizzare anche l'API ListUsers di Amazon Cognito, che accetta il parametro Filter.

Puoi cercare i seguenti attributi standard. Gli attributi personalizzati non possono essere cercati.

- username (distinzione tra maiuscole e minuscole)
- email
- phone_number
- name
- given_name
- family_name
- preferred_username
- cognito: user_status (denominato Status (Stato) nella console) (senza distinzione tra maiuscole e minuscole)
- status (denominato Enabled (Abilitato) nella console) (distinzione tra maiuscole e minuscole)
- sub

> **Note**
>
> È inoltre possibile fare un elenco degli utenti con un filtro lato client. Il filtro lato server non corrisponde a più di 1 attributo. Per la ricerca avanzata, utilizza un filtro lato client con il parametro `--query` dell'operazione `list-users` nella AWS Command Line Interface. Quando si utilizza un filtro lato client, ListUsers restituisce un elenco impaginato di zero o più utenti. È possibile ricevere più pagine di fila senza risultati. Ripetere la query con ogni token di impaginazione restituito fino a quando non si riceve un valore del token di impaginazione nullo, quindi rivedere il risultato combinato.

Per ulteriori informazioni sul filtro lato server e lato client, consulta Filtro dell'output di AWS CLI nella Guida per l'utente di AWS Command Line Interface (lingua italiana non garantita).

Ricerca di utenti tramite la AWS Management Console

Se hai già creato un bacino d'utenza, puoi effettuare la ricerca dal pannello Users (Utenti) nella AWS Management Console.

Le ricerche della AWS Management Console sono sempre ricerche con prefisso ("inizia con").

Per cercare un utente nella console Amazon Cognito

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Users (Utenti), quindi inserisci il nome utente nel campo di ricerca. Attenzione: alcuni valori di attributo tengono conto di maiuscole/minuscole (ad esempio lo User name (Nome utente).

   È inoltre possibile trovare utenti modificando il filtro di ricerca per restringere l'ambito alle altre proprietà dell'utente, ad esempio E-mail,numero di telefono, oppure cognome.

Ricerca di utenti tramite l'API **`ListUsers`**

Per cercare gli utenti dall'app, utilizza utilizza l'API ListUsers di Amazon Cognito. Quest'API utilizza i parametri seguenti:

- `AttributesToGet`: una matrice di stringhe, in cui ogni stringa è il nome di un attributo utente da restituire a ogni utente nei risultati di ricerca. Per recuperare tutti gli attributi, non includere un parametro `AttributesToGet` o richiesta `AttributesToGet` con un valore della stringa letterale `null`.
- `Filter`: una stringa del filtro del modulo"`AttributeName Filter-Type "AttributeValue""`. Le virgolette all'interno della stringa del filtro devono essere sfuggite utilizzando la barra rovesciata (\). Ad esempio, `"family_name = \"Reddy\""`. Se la stringa del filtro è vuota, `ListUsers` riporta tutti gli utenti al bacino d'utenza.
  - `AttributeName`: il nome dell'attributo da cercare. Puoi cercare solo un attributo per volta.

> **Note**
>
> Puoi cercare solo gli attributi standard. Gli attributi personalizzati non possono essere cercati. Questo perché si può effettuare la ricerca solo per gli attributi indicizzati, e gli attributi personalizzati non possono essere indicizzati.

  - `Filter-Type`: per una corrispondenza esatta, utilizza =, ad esempio, `given_name = "Jon"`. Per una corrispondenza del prefisso ("inizia con"), utilizza ^=, ad esempio, `given_name ^= "Jon"`.
  - `AttributeValue`: il valore di attributo che deve essere abbinato a ogni utente.
- `Limit`: numero massimo di utenti da restituire.
- `PaginationToken`: un token per ottenere più risultati da una ricerca precedente. Il token di impaginazione di Amazon Cognito scade dopo un'ora.
- `UserPoolId`: l'ID del bacino d'utenza per il bacino d'utenza sul quale eseguire la ricerca.

Tutte le ricerche non fanno la distinzione tra maiuscole e minuscole. I risultati di ricerca vengono ordinati in base all'attributo denominato dalla stringa `AttributeName`, in ordine crescente.

Esempi di utilizzo dell'API **`ListUsers`**

L'esempio seguente illustra la restituzione di tutti gli utenti e include tutti gli attributi.

```
{
    "AttributesToGet": null,
    "Filter": "",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

L'esempio seguente illustra la restituzione di tutti gli utenti il cui numero di telefono inizia per "+1312" e include tutti gli attributi.

```
{
    "AttributesToGet": null,
```

```
    "Filter": "phone_number ^= \"+1312\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

L'esempio seguente illustra la restituzione dei primi 10 utenti il cui cognome è "Reddy". Per ogni utente, i risultati di ricerca includono il nome specificato dall'utente, il numero di telefono e l'indirizzo e-mail. Se non ci sono più di 10 utenti che corrispondono all'interno del bacino d'utenza, la risposta include un token di paginazione.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "UserPoolId": "us-east-1_samplepool"
}
```

Mentre nell'esempio precedente viene restituito un token di paginazione, in quello seguente vengono restituiti i 10 utenti successivi che corrispondono alla stessa stringa del filtro.

```
{
    "AttributesToGet": [
        "given_name",
        "phone_number",
        "email"
    ],
    "Filter": "family_name = \"Reddy\"",
    "Limit": 10,
    "PaginationToken": "pagination_token_from_previous_search",
    "UserPoolId": "us-east-1_samplepool"
}
```

Recupero degli account utente

Il parametro `AccountRecoverySetting` consente di personalizzare il metodo che un utente può utilizzare per recuperare la password quando chiama l'API `ForgotPassword`. `ForgotPassword`

invia un codice di ripristino a un'e-mail verificata o a un numero di telefono verificato. Il codice di ripristino è valido per un'ora. Quando si specifica `AccountRecoverySetting` per il bacino d'utenza, Amazon Cognito sceglie la destinazione di distribuzione del codice in base alla priorità impostata.

Quando si definisce `AccountRecoverySetting` e un utente dispone di SMS MFA configurato, SMS non può essere utilizzato come meccanismo di recupero degli account. La priorità per questa impostazione è determinata con 1, la priorità più alta. Cognito invia una verifica a uno solo dei metodi specificati.

Ad esempio, `admin_only` è un valore utilizzato quando l'amministratore non desidera che l'utente recuperi autonomamente il proprio account e richieda invece di contattare l'amministratore per reimpostare il proprio account. Non è possibile utilizzare `admin_only` con altri meccanismi di recupero degli account.

Se non specifichi l'`AccountRecoverySetting`, Amazon Cognito utilizza il meccanismo legacy per determinare il metodo di recupero della password. In questo caso, Cognito utilizza prima un telefono verificato. Se il telefono verificato non viene trovato per l'utente, Cognito utilizzerà quindi l'e-mail verificata.

Per ulteriori informazioni su `AccountRecoverySetting`, consulta CreateUserPool e UpdateUserPool nella documentazione di riferimento dell'API del provider di identità di Amazon Cognito.

Come comportarsi in caso di password dimenticata

In una data ora, permettiamo tra 5 e 20 tentativi per un utente di richiedere o inserire un codice di reimpostazione della password come parte delle operazioni password-dimenticata e conferma-password-dimenticata. Il valore esatto dipende dai parametri di rischio associati alle richieste. Notare che questo comportamento è soggetto a modifiche.

Importazione di utenti in un bacino d'utenza

Sono disponibili due metodi per importare o migrare gli utenti dalla directory o dal database di utenti esistente nei bacini d'utenza di Amazon Cognito. Puoi eseguire la migrazione di utenti durante la procedura di accesso utilizzando Amazon Cognito per la prima volta con un trigger Lambda di migrazione utenti. Con questo approccio, gli utenti possono continuare a usare le loro password esistenti e non dovranno reimpostarle dopo la migrazione al bacino d'utenza. In alternativa, è possibile migrare gli utenti in blocco caricando un file CSV che contiene gli attributi del profilo utente di tutti gli utenti. Le seguenti sezioni descrivono questi due approcci.

Argomenti

- Importazione di utenti in bacini d'utenza con un trigger Lambda di migrazione utenti
- Importazione di utenti nel bacino d'utenza da un file CSV

Importazione di utenti in bacini d'utenza con un trigger Lambda di migrazione utenti

Questo approccio ti consente di eseguire senza problemi la migrazione degli utenti dalla directory degli utenti esistente ai bacini d'utenza, quando un utente accede per la prima volta con la tua app o richiede la reimpostazione della password. Aggiungi una funzione Trigger Lambda di migrazione utenti al bacino d'utenza per ricevere i metadati sugli utenti che tentano di accedere e ottenere le informazioni del profilo utente da un'origine di identità esterna. Per i dettagli e un codice di esempio su questo trigger Lambda, inclusi i parametri di richiesta e risposta, consulta Parametri del trigger Lambda di migrazione utenti.

Prima di avviare la migrazione degli utenti, crea una funzione Lambda di migrazione degli utenti nel tuo Account AWS e imposta la funzione Lambda come trigger di migrazione degli utenti nel bacino d'utenza. Aggiungi una policy di autorizzazione alla funzione Lambda che consenta solo al principale dell'account del servizio Amazon Cognito `cognito-idp.amazonaws.com` di richiamare la funzione Lambda e solo nel contesto del proprio bacino d'utenza. Per ulteriori informazioni, consulta Utilizzo delle policy basate su risorse per AWS Lambda (policy della funzione Lambda).

Processo di accesso

1. L'utente apre la tua app e accede con l'API dei bacini d'utenza di Amazon Cognito o tramite l'interfaccia utente ospitata di Amazon Cognito. Per ulteriori informazioni su come facilitare l'accesso con le API di Amazon Cognito, consulta Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili.
2. L'app invia il nome utente e la password ad Amazon Cognito. Se l'app ha un'interfaccia utente di accesso personalizzata creata con un AWSSDK, deve utilizzare InitiateAuth o AdminInitiateAuth con il flusso `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH`. Quando l'app utilizza uno di questi flussi, l'SDK invia la password al server.

> **Note**
>
> Prima di aggiungere un trigger di migrazione degli utenti, attiva il flusso `USER_PASSWORD_AUTH` o `ADMIN_USER_PASSWORD_AUTH` nelle impostazioni del client di app. Devi utilizzare questi flussi al posto del flusso `USER_SRP_AUTH` di default. Amazon

Cognito deve inviare una password alla funzione Lambda in modo che possa verificare l'autenticazione dell'utente nell'altra directory. Una SRP oscura la password dell'utente nella funzione Lambda.

3. Amazon Cognito verifica se il nome utente inviato corrisponde a un nome utente o un alias nel bacino d'utenza. Puoi impostare l'indirizzo e-mail, il numero di telefono o il nome utente preferito dell'utente come alias nel bacino d'utenza. Se l'utente non esiste, Amazon Cognito invia parametri, inclusi nome utente e password, alla funzione Trigger Lambda di migrazione utenti.
4. La funzione Trigger Lambda di migrazione utenti controlla o autentica l'utente nella directory o nel database degli utenti esistente. La funzione restituisce gli attributi utente archiviati da Amazon Cognito nel profilo dell'utente nel bacino d'utenza. Il parametro `username` viene restituito solo se il nome utente inviato corrisponde a un attributo alias. Se desideri che gli utenti continuino a usare le loro password esistenti, la funzione imposta l'attributo `finalUserStatus` su `CONFIRMED` nella risposta Lambda. L'app deve restituire tutti i parametri `"response"` mostrati in Parametri del trigger Lambda di migrazione utenti.

   Important

   Non registrare l'intero oggetto evento di richiesta nel codice Lambda di migrazione degli utenti in quanto questo oggetto evento di richiesta include la password dell'utente. Se i registri non sono sanificati, le password vengono visualizzate in CloudWatch Logs.

5. Amazon Cognito crea il profilo utente nel bacino d'utenza e restituisce i token per il client dell'app.
6. L'app esegue l'acquisizione del token, accetta l'autenticazione dell'utente e procede al contenuto richiesto.

Dopo aver eseguito la migrazione degli utenti, utilizza `USER_SRP_AUTH` per l'accesso. Il protocollo Secure Remote Password (SRP) non invia la password in rete e offre notevoli vantaggi dal punto di vista della sicurezza grazie al flusso `USER_PASSWORD_AUTH` usato durante la migrazione.

In caso di errori durante la migrazione, compresi i problemi del dispositivo client o di rete, l'app riceve le risposte di errore dall'API dei bacini d'utenza di Amazon Cognito. In questo caso, Amazon Cognito potrebbe non creare l'account utente nel bacino d'utenza. L'utente deve quindi tentare di eseguire nuovamente l'accesso. Se l'errore dell'accesso si ripete, tenta di reimpostare la password dell'utente con il flusso della password dimenticata dell'app.

Il flusso della password dimenticata richiama anche la funzione Trigger Lambda di migrazione utenti con un'origine eventi `UserMigration_ForgotPassword`. Poiché l'utente non invia una password quando richiede la reimpostazione della password, Amazon Cognito non include una password nell'evento che invia alla funzione Lambda. La funzione può solo cercare l'utente nella directory degli utenti esistente e restituire gli attributi da aggiungere al profilo utente nel bacino d'utenza. Dopo che la funzione ha completato l'invocazione e restituito la risposta ad Amazon Cognito, il pool di utenti invia un codice di reimpostazione della password tramite e-mail o SMS. Nell'app, richiedi all'utente di inserire il codice di conferma e una nuova password, quindi invia tali informazioni ad Amazon Cognito in una richiesta API ConfirmForgotPassword. Puoi anche utilizzare le pagine incorporate per il flusso della password dimenticata nell'interfaccia utente ospitata di Amazon Cognito.

Importazione di utenti nel bacino d'utenza da un file CSV

Puoi importare gli utenti in un bacino d'utenza di Amazon Cognito. Le informazioni degli utenti sono importate da un file .csv appositamente formattato. Il processo di importazione imposta i valori per tutti gli attributi utente eccetto la password. L'importazione della password non è supportata, poiché le best practice di sicurezza richiedono che le password non siano disponibili come testo normale e noi non supportiamo l'importazione di hash. Questo significa che i tuoi utenti devono cambiare le loro password dopo il primo accesso. Pertanto, quando si utilizza questo metodo per l'importazione, il loro stato sarà RESET_REQUIRED.

Puoi impostare le password degli utenti con una richiesta API AdminSetUserPassword che imposta il parametro `Permanent` su `true`.

> **Note**
>
> La data di creazione per ogni utente si riferisce al momento in cui tale utente è stato importato nel bacino d'utenza. La data di creazione non è uno degli attributi importati.

I passaggi di base sono:

1. Creare un ruolo Amazon CloudWatch Logs nella console AWS Identity and Access Management (IAM).
2. Crea il file .csv di importazione degli utenti;
3. Crea ed esegui Web;
4. Carica il file .csv di importazione degli utenti;

5. Avvia ed esegui Web;
6. Utilizza CloudWatch per controllare il log di eventi.
7. Richiedi agli utenti importati di ripristinare le loro password.

Argomenti

- Creazione del ruolo IAM CloudWatch Logs
- Creazione di un file CSV di importazione degli utenti
- Creazione ed esecuzione del processo di importazione del bacino d'utenza di Amazon Cognito
- Visualizzazione dei risultati dell'importazione del bacino d'utenza nella console CloudWatch
- Necessità degli utenti importati di ripristinare le password

Creazione del ruolo IAM CloudWatch Logs

Se utilizzi la CLI o l'API di Amazon Cognito, è necessario quindi creare un ruolo IAM per CloudWatch. Nella procedura seguente viene descritto come creare un ruolo IAM che può essere utilizzato da Amazon Cognito per scrivere i risultati del processo di importazione in CloudWatch Logs.

> **Note**
>
> Quando crei un processo di importazione nella console di Amazon Cognito, puoi creare il ruolo IAM contemporaneamente. Quando scegli l'operazione Create a new IAM role (Crea un nuovo ruolo IAM), Amazon Cognito applica automaticamente la policy di attendibilità e la policy IAM appropriate al ruolo.

Per creare il ruolo IAM di CloudWatch Logs per l'importazione del pool di utenti (AWS CLI, API)

1. Accedi alla AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Crea un nuovo ruolo IAM per un Servizio AWS. Per istruzioni dettagliate, consulta Creazione di un ruolo per un servizio Servizio AWS nella Guida per l'utente di AWS Identity and Access Management.

    a. Quando selezioni un Use Case (Caso d'uso) per Trusted entity type (Tipo di entità attendibile), scegli qualsiasi servizio. Amazon Cognito non è attualmente elencato nei casi d'uso dei servizi.

b. Nella schermata Add permissions (Aggiungi autorizzazioni), scegli Create policy (Crea policy) e inserisci la seguente istruzione della policy. Sostituisci *REGION* con la Regione AWS del pool di utenti, ad esempio `us-east-1`. Sostituisci *vpcID* con l'ID dell'Account AWS effettivo, ad esempio `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. Poiché non hai scelto Amazon Cognito come entità attendibile quando hai creato il ruolo, ora devi modificare manualmente la relazione di trust del ruolo. Seleziona Roles (Ruoli) nel pannello di navigazione della console IAM, quindi scegli il nuovo ruolo creato.
4. Seleziona la scheda Trust relationships (Relazioni di trust).
5. Seleziona Edit trust policy (Modifica policy di attendibilità).
6. Incolla la seguente istruzione della policy in Edit trust policy (Modifica policy di attendibilità), sostituendo qualsiasi testo esistente:

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "cognito-idp.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
```

```
            }
        ]
    }
```

7. Scegli Update policy (Aggiorna policy).
8. Nota il ruolo ARN. Quando si crea processo di importazione, è necessario fornire l'ARN.

Creazione di un file CSV di importazione degli utenti

Prima di poter importare gli utenti esistenti nel pool di utenti, è necessario creare un file CSV contenente gli utenti che si desidera importare e i relativi attributi. Dal pool di utenti, è possibile recuperare un file di importazione utente con intestazioni che riflettono lo schema di attributi del pool di utenti. È quindi possibile inserire le informazioni utente che soddisfano i requisiti di formattazione in Formattazione del file CSV.

Download dell'intestazione del file CSV (console)

Utilizza la seguente procedura per scaricare il file di intestazione CSV.

Per eseguire il download dell'intestazione del file CSV

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Users (Utenti);
5. Nella sezione Import users (Importa utenti), scegli Create an import job (Crea un processo di importazione).
6. In Upload CSV (Carica CSV), seleziona il link template.csv e scarica il file CSV.

Download dell'intestazione del file CSV (AWS CLI)

Per ottenere un elenco di intestazioni corrette, esegui i seguenti comandi della CLI, dove *USER_POOL_ID* (ID_DEL_POOL_DI_UTENTI) è l'identificatore del bacino d'utenza per il bacino d'utenza in cui importerai gli utenti:

```
aws cognito-idp get-csv-header --user-pool-id "USER_POOL_ID"
```

Risposta di esempio:

```
{
    "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"
}
```

Formattazione del file CSV

L'aspetto del file di intestazione CSV di importazione degli utenti scaricato è il seguente: Include anche eventuali attributi personalizzati aggiunti al pool di utenti.

```
cognito:username,name,given_name,family_name,middle_name,nickname,preferred_username,profile,pi
```

Modificare il file CSV, in modo che includa questa intestazione e i valori di attributo per gli utenti. Il file viene formattato in base ai seguenti criteri:

> **Note**
>
> Per ulteriori informazioni sui valori di attributo, ad esempio il formato corretto per i numeri di telefono, consulta Attributi del bacino d'utenza.

- La prima riga nel file è la riga di intestazione scaricata, che contiene i nomi di attributo degli utenti.
- L'ordine delle colonne nel file CSV non è rilevante.
- Ogni riga, dopo la prima, contiene i valori di attributo per un utente.
- Tutte le colonne nell'intestazione devono essere presenti, ma non è necessario che tu fornisca valori in ogni colonna;
- I seguenti attributi sono necessari:
  - cognito:username
  - cognito: mfa_enabled
  - email_verified o phone_number_verified
    - Almeno uno degli attributi verificato automaticamente deve essere `true` per ogni utente. Un attributo verificato automaticamente è un indirizzo e-mail o un numero di telefono a cui Amazon Cognito invia automaticamente un codice quando un nuovo utente si unisce al pool di utenti.
    - Il pool di utenti deve avere almeno un attributo verificato automaticamente, email_verified o phone_number_verified. Se il bacino d'utenza non ha attributi verificati automaticamente, il processo di importazione non verrà avviato.
    - Se il bacino d'utenza ha un attributo verificato automaticamente, tale attributo deve essere verificato per ogni utente. Ad esempio, se il bacino d'utenza ha solo phone_number come attributo verificato automaticamente, il valore phone_number_verified deve essere `true` per ogni utente.

> **Note**
>
> Affinché gli utenti reimpostino le password, devono avere una e-mail o un numero di telefono verificati. Amazon Cognito invia un messaggio contenente un codice di reimpostazione della password all'e-mail o al numero di telefono specificati nel file CSV. Se il messaggio viene inviato al numero di telefono, è inviato tramite SMS. Per

ulteriori informazioni, consulta Verifica delle informazioni di contatto al momento della registrazione.

  - e-mail (se email_verified è `true`)
  - phone_number (se phone_number_verified è `true`)
  - Tutti gli attributi che hai contrassegnato come richiesto quando hai creato il bacino d'utenza
- I valori di attributo che sono stringhe dovrebbero non essere tra virgolette;
- Se il valore di un attributo contiene una virgola, è necessario inserire una barra rovesciata (\) prima della virgola. Questo perché i campi in un file CSV sono separati da virgole.
- I contenuti del file CSV devono essere in formato UTF-8 senza contrassegno ordine di byte.
- Il campo cognito:username è obbligatorio e deve essere univoco all'interno del tuo bacino d'utenza. Può essere qualsiasi stringa Unicode. Tuttavia, non può contenere spazi o schede;
- I valori birthdate, se presenti, devono essere nel formato *`mm/gg/aaaa`*. Ciò significa, ad esempio, che una data di nascita corrispondente al 1 febbraio 1985 deve essere codificata come **`02/01/1985`**.
- Il campo cognito: mfa_enabled è obbligatorio. Se hai impostato l'autenticazione a più fattori o MFA (Multi-Factor Authentication) come obbligatoria nel tuo bacino d'utenza, questo campo deve essere `true` per tutti gli utenti. Se hai disattivato MFA, questo campo deve essere `false` per tutti gli utenti. Se hai impostato l'MFA come opzionale, questo campo può essere `true` o `false`, ma non può essere vuoto.
- La lunghezza massima è 16.000 caratteri.
- Il file CSV può avere una dimensione massima di 100 MB.
- Il numero massimo di righe (utenti) nel file è 500.000. In questo valore massimo non è inclusa la riga di intestazione.
- Il valore del campo updated_at dovrebbe essere un periodo di tempo espresso in secondi, per esempio: **`1471453471`**.
- Qualsiasi spazio, iniziale o finale in un valore di attributo, verrà tagliato.

L'elenco seguente è un esempio di file di importazione CSV per un pool di utenti senza attributi personalizzati. Lo schema del pool di utenti potrebbe differire da questo esempio. In tal caso, è necessario fornire valori di test nel modello CSV scaricato dal pool di utenti.

```
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
```

Creazione ed esecuzione del processo di importazione del bacino d'utenza di Amazon Cognito

In questa sezione viene descritto come creare ed eseguire il processo di importazione del pool di utenti utilizzando la console Amazon Cognito e AWS Command Line Interface (AWS CLI).

Argomenti

- Importazione di utenti da un file CSV (console)
- Importazione di utenti (AWS CLI)

Importazione di utenti da un file CSV (console)

Nella procedura seguente viene descritto come importare gli utenti dal file CSV.

Per importare gli utenti da un file CSV (console)

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS.
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Users (Utenti);
5. Nella sezione Import users (Importa utenti), scegli Create an import job (Crea un processo di importazione).
6. Nella pagina Create import job (Crea processo di importazione), inserisci un Job name (Nome processo).
7. Scegli l'operazione Create a new IAM role (Crea un nuovo ruolo IAM) o Use an existing IAM role (Utilizza un ruolo IAM esistente).

    a. Se hai scelto Create a new IAM role (Crea un nuovo ruolo IAM), inserisci un nome per il nuovo ruolo. Amazon Cognito creerà automaticamente un ruolo con le autorizzazioni e le relazioni di affidabilità corrette. Il principale IAM che crea il processo di importazione deve disporre delle autorizzazioni per creare ruoli IAM.

b. Se hai scelto Use an existing IAM role (Utilizza un ruolo IAM esistente), scegli un ruolo dall'elenco in IAM role selection (Selezione del ruolo IAM). Questo ruolo deve disporre delle autorizzazioni e della policy di attendibilità descritte in Creazione del ruolo IAM CloudWatch Logs.

8. Scegli Create job (Crea processo) per inviare il processo, ma avvialo in un secondo momento. Scegli Create and start job (Crea e avvia il processo) per inviare il processo e avviarlo immediatamente.
9. Se hai creato il processo ma non l'hai avviato, puoi avviarlo in un secondo momento. Nella scheda Users (Utenti) in Import users (Importa utenti), scegli il processo di importazione, quindi seleziona Start (Avvia). Puoi anche inviare una richiesta API StartUserImportJob da un SDK AWS.
10. Monitora lo stato di avanzamento del processo di importazione nella scheda Users (Utenti) in Import users (Importa utenti). Se il processo non va a buon fine, puoi selezionare il valore Status (Stato). Per ulteriori dettagli, seleziona View the CloudWatch logs for more details (Visualizza i log di CloudWatch per maggiori dettagli) ed esamina eventuali problemi nella console CloudWatch Logs.

Importazione di utenti (AWS CLI)

I seguenti comandi della CLI sono disponibili per importare gli utenti in un bacino d'utenza:

- `create-user-import-job`
- `get-csv-header`
- `describe-user-import-job`
- `list-user-import-jobs`
- `start-user-import-job`
- `stop-user-import-job`

Per l'elenco delle opzioni della riga di comando per questi comandi, utilizza l'opzione `help` della riga di comando. Ad esempio:

```
aws cognito-idp get-csv-header help
```

Creazione di un processo di importazione degli utenti

Dopo che hai creato il file CSV, crea un processo di importazione degli utenti eseguendo il comando della CLI seguente, dove *JOB_NAME* è il nome che scegli per il processo, *USER_POOL_ID* è l'ID del pool di utenti a cui verranno aggiunti i nuovi utenti, e *ROLE_ARN* è il ruolo ARN che hai ricevuto in Creazione del ruolo 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"
```

Il *PRE_SIGNED_URL* restituito nella risposta è valido per 15 minuti. Dopo questo lasso di tempo scadrà e, a quel punto, è necessario creare un processo di importazione degli utenti per ottenere un nuovo URL.

Example Risposta di esempio:

```
{
    "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
    }
}
```

Valori di stato per un processo di importazione degli utenti

Nelle risposte ai comandi di importazione degli utenti, visualizzerai uno dei seguenti valori `Status`:

- `Created`: Il processo è stato creato, ma non avviato.
- `Pending`: Stato di transizione. Hai avviato il processo, ma questo non ha ancora iniziato a importare gli utenti;
- `InProgress`: Il processo è stato avviato e gli utenti sono in fase di importazione.
- `Stopping`: Hai interrotto il processo, ma questo non ha ancora cessato l'importazione degli utenti.

- `Stopped`: Hai interrotto il processo e quest'ultimo ha cessato l'importazione degli utenti.
- `Succeeded`: La fase ha avuto esito positivo.
- `Failed`: Il processo è stato interrotto a causa di un errore.
- `Expired`: Hai creato un processo, ma non lo hai avviato entro 24-48 ore. Tutti i dati associati al processo sono stati eliminati e il processo non può essere avviato.

Caricamento del file CSV

Utilizza il seguente comando `curl` per caricare il file CSV contenente i dati dell'utente nell'URL prefirmato che hai ottenuto dalla risposta del comando `create-user-import-job`.

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

Nell'output di questo comando, cerca la frase "`We are completely uploaded and fine`". Questa frase indica che il file è stato caricato correttamente.

Descrizione di un processo di importazione degli utenti

Per una descrizione del processo di importazione degli utenti, utilizza il comando seguente, dove *`USER_POOL_ID`* è il tuo ID del bacino d'utenza e *`JOB_ID`* è l'ID del processo che è stato restituito quando hai creato il processo di importazione degli utenti.

```
aws cognito-idp describe-user-import-job --user-pool-id "USER_POOL_ID" --job-id
 "JOB_ID"
```

Example Risposta di esempio:

```
{
    "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
    }
}
```

Nel precedente output di esempio, *PRE_SIGNED_URL* è l'URL in cui hai caricato il file CSV. *ROLE_ARN* è l'ARN del ruolo di CloudWatch Logs ricevuto durante la creazione del ruolo.

Elenco dei tuoi processi di importazione degli utenti

Per elencare i tuoi processi di importazione degli utenti, utilizza il comando seguente:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 2
```

Example Risposta di esempio:

```
{
    "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"
}
```

I processi sono elencati in ordine cronologico dall'ultimo creato al primo. La stringa *PAGINATION_TOKEN* dopo il secondo processo indica che non ci sono risultati aggiuntivi per questo comando di elenco. Per elencare i risultati aggiuntivi, utilizza l' opzione `--pagination-token` come segue:

```
aws cognito-idp list-user-import-jobs --user-pool-id "USER_POOL_ID" --max-results 10 --
pagination-token "PAGINATION_TOKEN"
```

Avvio di un processo di importazione degli utenti

Per avviare un processo di importazione degli utenti, utilizza il comando seguente:

```
aws cognito-idp start-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Può essere attivo soltanto un processo di importazione alla volta per account.

Example Risposta di esempio:

```
{
    "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
    }
}
```

Arresto di un processo di importazione degli utenti

Per arrestare un processo di importazione degli utenti mentre è in corso, utilizza il comando seguente. Una volta che hai interrotto il processo, non puoi riavviarlo.

```
aws cognito-idp stop-user-import-job --user-pool-id "USER_POOL_ID" --job-id "JOB_ID"
```

Example Risposta di esempio:

```
{
    "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
    }
}
```

Visualizzazione dei risultati dell'importazione del bacino d'utenza nella console CloudWatch

Puoi visualizzare i risultati del tuo processo di importazione nella console Amazon CloudWatch.

Argomenti

- Visualizzazione dei risultati
- Interpretazione dei risultati

Visualizzazione dei risultati

I seguenti passaggi descrivono come visualizzare i risultati dell'importazione del bacino d'utenza.

Visualizzazione dei risultati dell'importazione del bacino d'utenza

1. Accedi alla AWS Management Console e apri la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Scegliere Logs (Log).
3. Scegli il gruppo di log per i tuoi processi di importazione del bacino d'utenza. Il nome del gruppo di log è del tipo ; `/aws/cognito/userpools/`*`USER_POOL_ID`*`/`*`USER_POOL_NAME`*.
4. Scegli il log per il processo di importazione degli utenti che hai appena eseguito. Il nome del gruppo di log è del tipo *`JOB_ID`*/*`JOB_NAME`*. I risultati nel log si riferiscono ai i tuoi utenti per numero di linea. Nessun dato degli utenti viene scritto nel log. Per ogni utente, viene visualizzata una linea simile a quella riportata di seguito:

   - `[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).`

Interpretazione dei risultati

Gli utenti importati correttamente hanno il loro stato impostato su "PasswordReset".

Nei seguenti casi, l'utente non verrà importato, ma il processo di importazione proseguirà:

- Nessun attributo verificato automaticamente è impostato su ; `true`.
- I dati dell'utente non corrispondono allo schema;
- Non è stato possibile importare l'utente a causa di un errore interno.

Nei seguenti casi, il processo di importazione fallirà:

- Il ruolo Amazon CloudWatch Logs non può essere assunto, non ha una policy d'accesso predefinita corretta o è stato eliminato.
- Il bacino d'utenza è stato eliminato;
- Amazon Cognito non è in grado di analizzare il file .csv.

Necessità degli utenti importati di ripristinare le password

La prima volta che ciascun utente importato esegue l'accesso e inserisce qualsiasi password, deve immettere una nuova password. Nella procedura seguente viene descritta l'esperienza utente in un'app personalizzata con utenti locali dopo l'importazione di un file CSV. Se gli utenti accedono con l'interfaccia utente ospitata, Amazon Cognito richiede di impostare una nuova password quando eseguono primo accesso.

Necessità degli utenti importati di ripristinare le password

1. Nell'app, tenta di accedere silenziosamente per l'utente corrente con `InitiateAuth` utilizzando una password casuale.
2. Amazon Cognito restituisce `NotAuthorizedException` quando `PreventUserExistenceErrors` è abilitato. In caso contrario, restituisce `PasswordResetRequiredException`.
3. L'app effettua una richiesta API `ForgotPassword` e reimposta la password dell'utente.

    a. L'app invia il nome utente in una richiesta API `ForgotPassword`.

    b. Amazon Cognito invia un codice all'indirizzo e-mail o al numero di telefono verificato. La destinazione dipende dai valori forniti per `email_verified` e `phone_number_verified` nel file CSV. La risposta alla richiesta `ForgotPassword` indica la destinazione del codice.

    > **Note**
    >
    > Il pool di utenti deve essere configurato per verificare le e-mail o i numeri di telefono. Per ulteriori informazioni, consulta Registrazione e conferma degli account utente.

    c. L'app visualizza un messaggio all'utente per verificare la posizione di invio del codice e richiede all'utente di inserire il codice e una nuova password.

    d. L'utente inserisce il codice e la nuova password nell'app;

    e. L'app invia il codice e la nuova password in una richiesta API `ConfirmForgotPassword`.

    f. L'app reindirizza l'utente per l'accesso.

Attributi del bacino d'utenza

Gli attributi sono informazioni che ti aiutano a identificare i singoli utenti, ad esempio nome, indirizzo e-mail e numero di telefono. Un nuovo bacino d'utenza include un set di attributi standard di

default. È inoltre possibile aggiungere attributi personalizzati alla definizione del pool di utenti in AWS Management Console. Questo argomento descrive questi attributi nel dettaglio e fornisce suggerimenti su come configurare il bacino d'utenza.

Non tutte le informazioni sugli utenti devono essere archiviate negli attributi. Ad esempio, i dati utente che cambiano spesso, come le statistiche di utilizzo o i punteggi dei giochi, devono essere conservati in un archivio dati separato, come Amazon Cognito Sync o Amazon DynamoDB.

> **Note**
>
> In alcuni documenti e standard si fa riferimento agli attributi come membri.

Argomenti

- Standard attributes (Attributi standard)
- Nomi utente e nomi utente preferiti
- Personalizzazione degli attributi di accesso
- Attributi personalizzati
- Autorizzazioni attributo e ambiti

Standard attributes (Attributi standard)

Amazon Cognito assegna a tutti gli utenti un set di attributi standard in base alla specifica OpenID Connect. Per impostazione predefinita, i valori degli attributi standard e personalizzati possono essere costituiti da qualsiasi stringa composta da un massimo di 2.048 caratteri, ma alcuni valori di attributo hanno restrizioni di formato.

Gli attributi standard sono:

- `address`
- `birthdate`
- `email`
- `family_name`
- `gender`
- `given_name`
- `locale`

- middle_name
- name
- nickname
- phone_number
- picture
- preferred_username
- profile
- sub
- updated_at
- website
- zoneinfo

Ad eccezione di sub, per impostazione predefinita, gli attributi standard sono opzionali per tutti gli utenti. Per rendere obbligatorio un attributo, durante il processo di creazione del bacino d'utenza, spunta la casella di controllo Required (obbligatorio) accanto all'attributo. Amazon Cognito assegna un valore identificativo utente unico all'attributo sub di ciascun utente. È possibile verificare solo gli attributi email e phone_number.

> **Note**
>
> Quando contrassegni un attributo standard come Required (Obbligatorio), un utente non può effettuare la registrazione, a meno che non fornisca un valore per l'attributo. Per creare utenti e non fornire valori per gli attributi obbligatori, gli amministratori possono utilizzare l'AdminCreateUserAPI. Dopo aver creato il bacino d'utenza, non è più possibile cambiare un attributo da obbligatorio a non obbligatorio e viceversa.

Dettagli degli attributi standard e restrizioni di formato

birthdate

Il valore deve essere una data valida di 10 caratteri nel formato AAAA-MM-DD.

e-mail

Utenti e amministratori possono verificare i valori dell'indirizzo e-mail.

Un amministratore con Account AWS le autorizzazioni appropriate può modificare l'indirizzo e-mail dell'utente e contrassegnarlo come verificato. Contrassegna un indirizzo email come verificato con l'AdminUpdateUserAttributesAPI o il comando admin-update-user-attributes AWS Command Line Interface (AWS CLI). Con questo comando, l'amministratore può impostare l'attributo `email_verified` su `true`. Puoi anche modificare un utente nella scheda Utenti di AWS Management Console per contrassegnare un indirizzo email come verificato.

Il valore deve essere una stringa di indirizzo e-mail valida conforme al formato e-mail standard con il simbolo @ e il dominio, con una lunghezza massima di 2.048 caratteri.

phone_number

Un utente deve fornire un numero di telefono se è attiva l'autenticazione a più fattori (MFA) con SMS. Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.

Utenti e amministratori possono verificare i valori del numero di telefono.

Un amministratore con Account AWS le autorizzazioni appropriate può modificare il numero di telefono dell'utente e contrassegnarlo come verificato. Contrassegna un numero di telefono come verificato con l'AdminUpdateUserAttributesAPI o il admin-update-user-attributes AWS CLI comando. Con questo comando, l'amministratore può impostare l'attributo `phone_number_verified` su `true`. Puoi anche modificare un utente nella scheda Utenti di AWS Management Console per contrassegnare un numero di telefono come verificato.

> **Important**
>
> I numeri di telefono devono attenersi alle seguenti regole di formattazione: devono iniziare con un segno più (**+**), seguito immediatamente dal codice del paese. Un numero di telefono può contenere solo il segno **+** e le cifre. Prima di inviare il valore al servizio, rimuovi dal numero di telefono qualsiasi altro carattere, come parentesi, spazi o trattini (**-**). Un numero di telefono degli Stati Uniti ad esempio, deve seguire questo formato: **`+14325551212`**.

preferred_username

È possibile selezionare `preferred_username` come obbligatorio o come alias, ma non entrambi. Se `preferred_username` è un alias, puoi fare una richiesta all'operazione UpdateUserAttributesAPI e aggiungere il valore dell'attributo dopo aver confermato l'utente.

sub

Indicizza e cerca i tuoi utenti in base all'attributo `sub`. L'attributo `sub` è un identificativo utente unico all'interno di ciascun pool di utenti. Gli utenti possono modificare attributi come `phone_number` e `email`. L'attributo `sub` ha un valore fisso. Per ulteriori informazioni sull'esito di utenti, consultare Gestione e ricerca degli account utente.

Visualizza gli attributi obbligatori

Utilizza la procedura seguente per visualizzare gli attributi obbligatori per un determinato bacino d'utenza.

> **Note**
>
> Non è possibile modificare gli attributi obbligatori dopo che il bacino d'utenza è stato creato.

Visualizzare gli attributi obbligatori

1. Vai ad Amazon Cognito in. AWS Management Console Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Seleziona un bacino d'utenza esistente dall'elenco.
4. Seleziona la scheda Sign-in experience (esperienza di accesso).
5. Visualizza gli attributi obbligatori del bacino d'utenza nella sezione Required attributes (Attributi obbligatori).

Nomi utente e nomi utente preferiti

Il valore `username` è un attributo separato ed è diverso dall'attributo `name`. Ogni utente ha un attributo `username`. Amazon Cognito genera automaticamente un nome utente per gli utenti federati. È necessario fornire un attributo `username` per creare un utente locale nella directory Amazon Cognito. Dopo aver creato un utente, non puoi più modificare il valore dell'attributo `username`.

Gli sviluppatori possono utilizzare l'attributo `preferred_username` per fornire agli utenti un nome utente che possano modificare. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Se l'applicazione non richiede un nome utente, non è necessario chiedere agli utenti di fornirne uno. L'app è in grado di creare un nome utente univoco per gli utenti in background. Questa funzione è utile se, ad esempio, desideri che gli utenti si registrino e accedano con un indirizzo e-mail e una password. Per ulteriori informazioni, consulta Personalizzazione degli attributi di accesso.

Il `username` deve essere univoco all'interno di un bacino d'utenza. Un `username` può essere riutilizzato, ma solo dopo che è stato eliminato e non è più in uso. Per informazioni sui vincoli di stringa agli `username` attributi, consulta la proprietà username di una richiesta API. SignUp

Personalizzazione degli attributi di accesso

Quando crei un pool di utenti, puoi configurare attributi del nome utente se gli utenti devono essere in grado di effettuare la registrazione e l'accesso utilizzando un indirizzo e-mail o un numero di telefono come il rispettivo nome utente. In alternativa, è possibile configurare attributi alias per fornire agli utenti la possibilità di includere più attributi quando effettuano la registrazione e quindi l'accesso con un nome utente, un nome utente preferito, un indirizzo e-mail o un numero di telefono.

> **Important**
>
> Dopo aver creato il bacino d'utenza, non puoi più cambiare questa impostazione.

Come scegliere tra attributi alias e attributi nome utente

Il tuo requisito	Attributi alias	Attributi nome utente
Gli utenti dispongono di più attributi di accesso	Sì[1]	No[2]
Gli utenti devono verificare l'indirizzo e-mail o il numero di telefono prima di poter accedere con esso	Sì	No
Registra gli utenti con indirizzi e-mail o numeri di telefono duplicati e previeni `UsernameExistsException` gli errori[3]	Sì	No

Il tuo requisito	Attributi alias	Attributi nome utente
Può assegnare lo stesso valore di attributo dell'indirizzo e-mail o del numero di telefono a più di un utente	Sì[4]	No

[1] Gli attributi di accesso disponibili sono nome utente, indirizzo e-mail, numero di telefono e nome utente preferito.

[2] Può accedere utilizzando l'indirizzo e-mail o il numero di telefono.

[3] Il tuo pool di utenti non genera errori `UsernameExistsException` quando gli utenti si registrano con indirizzi e-mail o numeri di telefono potenzialmente duplicati, ma senza nome utente. Questo comportamento è indipendente da Previeni errori di presenza degli utenti, che si applica alle operazioni di accesso, ma non di registrazione.

[4] Solo l'ultimo utente che ha verificato l'attributo può utilizzarlo per l'accesso.

Opzione 1: attributi di accesso multipli (attributi alias)

Se desideri consentire agli utenti di scegliere di inserire il nome utente o altri valori di attributo quando effettuano l'accesso, puoi attivare gli alias. Per impostazione predefinita, gli utenti accedono con il proprio nome utente e password. Il nome utente è un valore fisso che gli utenti non possono modificare. Se contrassegni un attributo come alias, gli utenti possono effettuare l'accesso usando l'attributo al posto del nome utente. Puoi contrassegnare come alias gli attributi dell'indirizzo e-mail, del numero di telefono e del nome utente preferito. Ad esempio, se contrassegni l'e-mail e il numero telefono come alias per un pool di utenti, gli utenti di quel pool di utenti possono effettuare l'accesso utilizzando il proprio nome utente, indirizzo e-mail o numero di telefono, insieme alla password.

Per scegliere gli attributi alias, seleziona User name (Nome utente) e almeno un'opzione di accesso aggiuntiva quando crei il pool di utenti.

> **Note**
>
> Se il bacino d'utenza è configurato per non applicare la distinzione tra maiuscole e minuscole, un utente può utilizzare lettere minuscole o maiuscole per registrarsi o accedere con l'alias.

Per ulteriori informazioni, consulta il riferimento CreateUserPoolall'API dei pool di utenti di Amazon Cognito.

Se selezioni l'indirizzo e-mail come alias, Amazon Cognito non accetta un nome utente corrispondente a un formato di indirizzo e-mail valido. Analogamente, se selezioni un numero di telefono come alias, Amazon Cognito non accetta per il pool di utenti un nome utente che corrisponde a un formato di numero di telefono valido.

> **Note**
>
> I valori alias devono essere univoci in un bacino d'utenza. Se configuri un alias per un indirizzo e-mail o un numero di telefono, il valore che fornisci può essere nello stato verificato in un solo account. Se durante la registrazione l'utente fornisce un indirizzo e-mail o un numero di telefono come valore alias e un altro utente ha già utilizzato quel valore alias, la registrazione viene completata. Tuttavia, quando un utente cerca di confermare l'account con l'indirizzo e-mail (o il numero di telefono) e immette il codice valido, Amazon Cognito restituisce un errore `AliasExistsException`. L'errore indica all'utente che un account con l'indirizzo e-mail (o il numero di telefono) specificato è già esistente. A questo punto, l'utente può abbandonare la creazione del nuovo account e tentare di ripristinare la password dell'account precedente. Se l'utente continua a creare il nuovo account, l'app deve chiamare l'API `ConfirmSignUp` con l'opzione `forceAliasCreation`. `ConfirmSignUp` con `forceAliasCreation` sposta l'alias dall'account precedente all'account appena creato e contrassegna l'attributo come non verificato nell'account precedente.

I numeri di telefono e gli indirizzi e-mail diventano alias attivi per un utente solo dopo che vengono verificati dall'utente. Ti consigliamo di scegliere la verifica automatica degli indirizzi e-mail e dei numeri di telefono se li utilizzi come alias.

Scegli gli attributi alias per evitare errori `UsernameExistsException` negli attributi dell'indirizzo e-mail e del numero di telefono quando gli utenti effettuano la registrazione.

Attiva l'attributo `preferred_username` in modo che l'utente possa modificare il nome utente che utilizza per accedere quando il valore dell'attributo `username` non cambia. Se desideri configurare questa esperienza utente, invia il nuovo valore `username` come `preferred_username` e scegli il `preferred_username` come alias. Gli utenti possono in questo modo effettuare l'accesso

inserendo il nuovo valore. Se selezioni `preferred_username` come alias, gli utenti possono fornire il valore solo quando confermano l'account e non durante la registrazione.

Quando l'utente effettua la registrazione con un nome utente, puoi scegliere se può accedere con uno o più dei seguenti alias.

- Un indirizzo e-mail verificato
- Un numero di telefono verificato
- Il nome utente preferito

Questi alias possono essere modificati dall'utente dopo che ha effettuato la registrazione.

> **Important**
>
> Se il tuo pool di utenti supporta l'accesso con alias e desideri autorizzare o cercare un utente, non identificarlo in base a nessuno dei suoi attributi di accesso. L'identificatore utente a valore fisso `sub` è l'unico indicatore coerente dell'identità dell'utente.

Includi i passaggi seguenti quando crei il bacino d'utenza in modo che gli utenti possano accedere con un alias.

Configurazione di un pool di utenti per accedere con un nome utente preferito

1. Passa ad Amazon Cognito nella AWS Management Console. Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Nell'angolo in alto a destra della pagina, seleziona Create a User Pool (Crea bacino d'utenza) per avviare la procedura guidata di creazione del bacino d'utenza.
4. Alla voce Configure sign-in experience (configura l'esperienza di accesso), seleziona i tipi di provider di identità che desideri associare al bacino d'utenza.
5. Alla voce Cognito user pool sign-in options (opzioni di accesso al pool di utenti Cognito), scegli qualsiasi combinazione dinome utente,E-mail, e Numero di telefono.
6. Per Requisiti per i nomi utente, seleziona Consenti agli utenti di accedere con un nome utente preferito per consentire agli utenti di impostare un nome utente alternativo al momento dell'accesso.

7. Seleziona Next step (Successivo) per salvare e completare tutti i passaggi nella procedura guidata.

Opzione 2: indirizzo e-mail o numero di telefono come attributo di accesso (attributi nome utente)

Quando l'utente si registra con un indirizzo e-mail o un numero di telefono come nome utente, puoi scegliere se eseguire la registrazione solo con l'indirizzo e-mail, solo con il numero di telefono o con uno dei due a scelta.

Per scegliere gli attributi nome utente, non selezionare Nome utente come opzione di accesso quando crei il pool di utenti.

L'indirizzo e-mail o il numero di telefono devono essere univoci e non essere già in uso per un altro utente. Non è necessario che siano verificati. Dopo che l'utente ha effettuato la registrazione utilizzando un indirizzo e-mail o un numero di telefono, non può creare un nuovo account con lo stesso indirizzo e-mail o numero di telefono. L'utente può solo riutilizzare l'account esistente e reimpostare la password dell'account, se necessario. Tuttavia, può modificare l'indirizzo e-mail o il numero di telefono con un nuovo indirizzo e-mail o numero di telefono. Se l'indirizzo e-mail o il numero di telefono non sono già in uso diventano il nuovo nome utente.

> **Note**
>
> Quando un utente effettua la registrazione utilizzando un indirizzo e-mail come proprio nome utente, può sostituire il nome utente con un altro indirizzo e-mail, ma non con un numero di telefono. Allo stesso modo, se effettua la registrazione utilizzando un numero di telefono come proprio nome utente, può sostituire il nome utente con un altro numero di telefono, ma non con un indirizzo e-mail.

Utilizza i passaggi seguenti durante il processo di creazione del bacino d'utenza per configurare la registrazione e l'accesso con l'indirizzo e-mail o il numero di telefono.

Configurazione di un bacino d'utenza per la registrazione e l'accesso utilizzando un indirizzo e-mail o un numero di telefono

1. Passa ad Amazon Cognito nella AWS Management Console. Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).

3. Nell'angolo in alto a destra della pagina, scegli Create a User Pool (Crea bacino d'utenza).
4. In Cognito user pool sign-in options (Opzioni di accesso al pool di utenti Cognito), seleziona qualsiasi combinazione di Email (Indirizzo e-mail) e Phone number (Numero di telefono) che rappresenta gli attributi alias che l'utente può usare per eseguire l'accesso.
5. Seleziona Next (Successivo) e completa tutti i passaggi rimanenti nella procedura guidata.

> **Note**
>
> Non è necessario contrassegnare l'indirizzo e-mail o il numero di telefono come attributi obbligatori per il bacino d'utenza.

Implementazione dell'opzione 2 nell'app

1. Chiama l'API `CreateUserPool` per creare il bacino d'utenza. Imposta il parametro `UserNameAttributes` su `phone_number`, `email` o `phone_number | email`.
2. Chiama l'API `SignUp` e passa un indirizzo e-mail o un numero di telefono nel parametro `username` dell'API. Quest'API svolge le funzioni seguenti:
   - Se la stringa `username` è in un formato di indirizzo e-mail valido, il bacino d'utenza popola automaticamente l'attributo `email` dell'utente con il valore `username`.
   - Se la stringa `username` è in un formato di numero di telefono valido, il bacino d'utenza popola automaticamente l'attributo `phone_number` dell'utente con il valore `username`.
   - Se la stringa `username` non è in un formato di indirizzo e-mail o numero di telefono, l'API `SignUp` restituisce un'eccezione.
   - L'API `SignUp` genera un UUID persistente per l'utente e lo utilizza internamente come attributo del nome utente immutabile. Questo UUID ha lo stesso valore dell'attestazione `sub` nel token di identità dell'utente.
   - Se la stringa `username` contiene un indirizzo e-mail o un numero di telefono già in uso, l'API `SignUp` restituisce un'eccezione.

Puoi utilizzare un indirizzo e-mail o numero di telefono come alias al posto del nome utente in tutte le API, tranne l'API `ListUsers`. Quando chiami `ListUsers`, puoi effettuare la ricerca in base all'attributo `email` o `phone_number`. Se esegui una ricerca per `username`, devi fornire il nome utente effettivo e non un alias.

Attributi personalizzati

È possibile aggiungere fino a 50 attributi personalizzati al bacino d'utenza. Per gli attributi personalizzati, si può specificare una lunghezza minima e/o massima. Tuttavia, la lunghezza massima per qualsiasi attributo personalizzato non può superare i 2048 caratteri.

Ogni attributo personalizzato presenta le caratteristiche seguenti:

- Può essere definito come stringa o numero. Amazon Cognito scrive i valori degli attributi personalizzati nel token ID solo come stringhe.
- Non è possibile richiedere che gli utenti forniscano un valore per l'attributo.
- Non è possibile eliminarlo o modificarlo dopo averlo aggiunto al bacino d'utenza.
- La lunghezza in caratteri del nome dell'attributo rientra nel limite accettato da Amazon Cognito. Per ulteriori informazioni, consulta Quote in Amazon Cognito.
- Può essere mutabile o immutabile. Puoi scrivere un valore in un attributo immutabile solo quando crei un utente. Puoi modificare il valore di un attributo mutabile se il client dell'app dispone dell'autorizzazione di scrittura per l'attributo. Per ulteriori informazioni, consulta Autorizzazioni attributo e ambiti.

> **Note**
>
> Nel codice e nelle impostazioni delle regole per Utilizzo del controllo degli accessi basato su ruoli, gli attributi personalizzati richiedono il prefisso `custom:` per distinguerli dagli attributi standard.

Puoi anche aggiungere attributi di sviluppatore quando crei pool di utenti, nella `SchemaAttributes` proprietà di. CreateUserPool Gli attributi di sviluppatore hanno un prefisso `dev:`. È possibile modificare gli attributi di sviluppatore di un utente solo con AWS credenziali. Gli attributi di sviluppatore sono una funzionalità legacy che è stata sostituita da Amazon Cognito con le autorizzazioni di lettura/scrittura del client dell'app.

Utilizza la procedura seguente per creare un attributo delle chiavi personalizzate.

Aggiunta di un attributo personalizzato utilizzando la console

1. Vai ad Amazon Cognito in. AWS Management Console Se la console te lo richiede, inserisci le tue credenziali. AWS

2. Scegli User Pools (bacini d'utenza).
3. Seleziona un bacino d'utenza esistente dall'elenco.
4. Seleziona la scheda Sign-up experience (esperienza di registrazione), e nella sezione Custom attributes (attributi personalizzati), seleziona Add custom attributes (aggiunta di attributi personalizzati).
5. Nella pagina Add custom attributes (aggiungi attributi personalizzati) fornisci i seguenti dettagli sul nuovo attributo:
   - Immetti un nome.
   - Seleziona un tipo tra String o Number.
   - Immetti un valore numerico o una lunghezza minima della stringa.
   - Immetti un valore numerico o una lunghezza massima della stringa.
   - Seleziona Mutable (Mutabile) se desideri concedere agli utenti l'autorizzazione per modificare il valore di un attributo personalizzato dopo l'impostazione del valore iniziale.
6. Seleziona Salvataggio delle modifiche.

Autorizzazioni attributo e ambiti

Per ogni client di app puoi impostare le autorizzazioni di lettura e scrittura di ogni attributo utente. In questo modo, puoi controllare l'accesso concesso a qualsiasi app per leggere e modificare ogni attributo che archivi per gli utenti. Ad esempio, puoi avere un attributo personalizzato che indica se un utente è un cliente pagante o meno. Le tue app possono essere in grado di vedere questo attributo, ma non di modificarlo direttamente. Al contrario, puoi aggiornare questo attributo utilizzando uno strumento di amministrazione o un processo in background. Puoi impostare le autorizzazioni per gli attributi utente con la console Amazon Cognito, l'API Amazon Cognito o AWS CLI. Di default, tutti i nuovi attributi personalizzati saranno disponibili solo dopo aver impostato le relative autorizzazioni di lettura e scrittura. Per impostazione predefinita, quando crei un nuovo client per l'app, concedi all'app le autorizzazioni di lettura e scrittura per tutti gli attributi standard e personalizzati. Per limitare l'app alla sola quantità di informazioni richiesta, assegna autorizzazioni specifiche agli attributi nella configurazione del client dell'app.

Come procedura ottimale, specifica le autorizzazioni di lettura e scrittura degli attributi quando crei un client per l'app. Concedi al client dell'app l'accesso al set minimo di attributi utente necessari per il funzionamento dell'applicazione.

> **Note**
>
> DescribeUserPoolClientrestituisce valori solo per `ReadAttributes` e `WriteAttributes` quando configuri autorizzazioni del client dell'app diverse da quelle predefinite.

Come aggiornare le autorizzazioni degli attributi (AWS Management Console)

1. Vai ad Amazon Cognito in. AWS Management Console Se la console te lo richiede, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Seleziona un bacino d'utenza esistente dall'elenco.
4. Seleziona la scheda App integration (integrazione App), e nella sezione App clients (client dell'App) seleziona un client di App dall'elenco.
5. Nella sezione Attribute read and write permissions (autorizzazioni di lettura e scrittura degli attributi), seleziona l'opzione Edit (modifica).
6. Nella pagina Edit attribute read and write permissions (impostazione delle autorizzazioni di lettura e scrittura degli attributi) configura le autorizzazioni di lettura e scrittura, quindi seleziona Save changes (salva modifiche).

Ripeti questi passaggi per ogni client di app che usa l'attributo personalizzato.

Per ogni app, puoi contrassegnare gli attributi come leggibili o scrivibili. Questo vale sia per gli attributi standard sia per quelli personalizzati. La tua app può recuperare il valore degli attributi contrassegnati come leggibili e può impostare o modificare il valore degli attributi contrassegnati come scrivibili. Se la tua app tenta di impostare un valore per un attributo che non è autorizzata a scrivere, Amazon Cognito restituisce `NotAuthorizedException`. GetUserle richieste includono un token di accesso con una dichiarazione del client dell'app; Amazon Cognito restituisce solo valori per gli attributi che il client dell'app è in grado di leggere. Il token ID dell'utente di un'app contiene solo le attestazioni che corrispondono agli attributi leggibili. Tutti i client dell'app possono scrivere gli attributi richiesti dal pool di utenti. Puoi impostare il valore di un attributo in una richiesta API del pool di utenti di Amazon Cognito solo quando fornisci anche un valore per gli attributi obbligatori che non hanno ancora un valore.

Gli attributi personalizzati dispongono di funzionalità distinte per le autorizzazioni di lettura e scrittura. È possibile creare questi attributi come mutabili o immutabili per il pool di utenti e impostarli come attributi di lettura o scrittura per qualsiasi client dell'app.

Un attributo personalizzato immutabile può essere aggiornato una sola volta, durante la creazione dell'utente. È possibile popolare un attributo immutabile con i seguenti metodi.

- `SignUp`: un utente esegue la registrazione mediante un client dell'app che dispone dell'accesso in scrittura a un attributo personalizzato immutabile. Specifica un valore per tale attributo.
- Accesso con un gestore dell'identità digitale di terze parti: un utente accede a un client dell'app con accesso in scrittura a un attributo personalizzato immutabile. La configurazione del pool di utenti per il relativo gestore dell'identità digitale prevede una regola per mappare una richiesta specificata a un attributo immutabile.
- `AdminCreateUser`: viene fornito un valore per un attributo immutabile.

Per informazioni sugli ambiti che puoi assegnare ai client dell'app, consulta Autorizzazione Scopes, M2M e API con server di risorse.

Puoi modificare le autorizzazioni e gli ambiti attributo dopo aver creato il bacino d'utenza.

Aggiunta di requisiti password del bacino d'utenza

Le password complesse e sicure sono una best practice di sicurezza per il tuo pool di utenti. Soprattutto nelle applicazioni aperte a Internet, le password deboli possono esporre le credenziali degli utenti a sistemi che indovinano le password e cercano di accedere ai dati. Più una password è complessa, più è difficile indovinarla. Amazon Cognito offre strumenti aggiuntivi per gli amministratori attenti alla sicurezza, come funzionalità di sicurezza avanzate e ACL AWS WAF Web, ma la politica delle password è un elemento centrale della sicurezza della directory utenti.

Le password per gli utenti locali nei pool di utenti di Amazon Cognito non scadono automaticamente. Come best practice, registra l'ora, la data e i metadati delle reimpostazioni delle password degli utenti in un sistema esterno. Con un registro esterno dell'età della password, l'applicazione o un trigger Lambda possono cercare l'età della password di un utente e richiederne la reimpostazione dopo un determinato periodo.

Puoi configurare il tuo pool di utenti in modo che richieda una complessità minima della password conforme ai tuoi standard di sicurezza. Le password complesse hanno una lunghezza minima di almeno otto caratteri. Includono anche una combinazione di caratteri maiuscoli, numerici e speciali.

Impostazione di una policy delle password di un pool di utenti

1. Crea un pool di utenti e vai alla fase Configurazione dei requisiti di sicurezza oppure accedi a un pool di utenti esistente e vai alla scheda Esperienza di accesso.

2. Vai a Policy delle password.
3. Scegli una Modalità policy delle password. I Valori predefiniti di Cognito configurano il pool di utenti con le impostazioni minime consigliate. Puoi anche scegliere una policy delle password Personalizzata.
4. Imposta una Lunghezza minima della password. Tutti gli utenti devono registrarsi o essere creati con una password la cui lunghezza sia maggiore o uguale a questo valore. Puoi impostare questo valore minimo fino a 99, ma gli utenti possono impostare password lunghe fino a 256 caratteri.
5. Configura le regole di complessità delle password in Requisiti delle password. Scegli i tipi di caratteri (numeri, caratteri speciali, lettere maiuscole e minuscole) che devono essere presenti almeno in una occorrenza nelle password di ogni utente.

   È possibile richiedere almeno uno dei seguenti caratteri nelle password. Dopo che Amazon Cognito ha verificato che le password contengano i caratteri minimi richiesti, le password degli utenti possono contenere caratteri aggiuntivi di qualsiasi tipo fino alla lunghezza massima della password.

   - Lettere maiuscole e minuscole dell'alfabeto latino di base
   - Numeri
   - Possono includere i seguenti caratteri speciali.

     ```
     ^ $ * . [ ] { } ( ) ? " ! @ # % & / \ , > < ' : ; | _ ~ ` = + -
     ```

   - Spazi non iniziali, non finali.
6. Imposta un valore per Scadenza delle password temporanee impostate dagli amministratori. Trascorso questo periodo di tempo, un nuovo utente creato con una richiesta API `AdminCreateUser` nella console di Amazon Cognito non può effettuare l'accesso e impostare una nuova password. Dopo aver effettuato l'accesso con la password temporanea, i relativi account utente non scadono mai. Per aggiornare la durata della password nell'API dei pool di utenti di Amazon Cognito, imposta un valore per TemporaryPasswordValidityDays nella tua richiesta CreateUserPoolo UpdateUserPoolAPI.

   - Per reimpostare l'accesso per un account utente scaduto, effettua una delle seguenti operazioni.
     - Elimina la CA e creane una nuova.
     - Imposta una nuova password permanente in una richiesta AdminSetUserPasswordAPI.

- Genera un nuovo codice di conferma in una richiesta AdminResetUserPasswordAPI.

Impostazioni e-mail per i bacini d'utenza di Amazon Cognito

Alcuni eventi nell'App client del bacino d'utenza potrebbero causare l'invio di e-mail agli utenti da parte di Amazon Cognito. Ad esempio, se configuri il bacino d'utenza in modo da richiedere la verifica e-mail, Amazon Cognito invia un'e-mail quando un utente accede a un nuovo account nell'app o quando reimposta la password. A seconda dell'azione che attiva l'e-mail, l'e-mail contiene un codice di verifica o una password temporanea.

Per gestire l'invio delle e-mail, è possibile utilizzare le seguenti opzioni:

- La configurazione e-mail predefinita integrata nel servizio Amazon Cognito.
- La configurazione di Amazon Simple Email Service (Amazon SES).

Puoi modificare l'opzione di distribuzione dopo aver creato il pool di utenti.

Amazon Cognito invia messaggi e-mail agli utenti con un codice che possono inserire o un collegamento URL che possono selezionare. Nella tabella seguente vengono mostrati gli eventi che possono generare un messaggio di posta elettronica.

Opzioni relative ai messaggi

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Password dimenticata	ForgotPassword	E-mail, SMS	code	No	N/D
Invito	AdminCreateUser	Posta elettronica, SMS	code	Sì	Messaggio di invito
Autoregistrazione	SignUp	E-mail, SMS	codice, link	Sì	messaggio di verifica
Verifica dell'indirizzo	UpdateUserAttributes	E-mail, SMS	code	Sì	Messaggio di verifica

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
e-mail o del numero di telefono					
Autenticazione a più fattori (MFA)	AdminInitiateAuth, InitiateAuth	SMS	code	Sì[1]	Messaggio MFA

[1] Per messaggi SMS.

Amazon SES addebita il costo dei messaggi e-mail. Per ulteriori informazioni, consulta la pagina dei Prezzi di Amazon SES.

Configurazione e-mail predefinita

Amazon Cognito può utilizzare la sua configurazione e-mail predefinita per gestire le consegne di posta elettronica per te. Quando utilizzi l'opzione di default, Amazon Cognito limita il numero di e-mail che puoi inviare al giorno al tuo bacino d'utenza. Per ulteriori informazioni sui limiti del servizio, consulta Quote in Amazon Cognito. Per gli ambienti di produzione tipici, il limite di e-mail predefinito è inferiore al volume di distribuzione richiesto. Per consentire un volume di consegna più alto, è possibile utilizzare la configurazione e-mail di Amazon SES.

Quando si utilizza la configurazione predefinita, si utilizzano risorse Amazon SES gestite da AWS per inviare messaggi e-mail. Amazon SES aggiunge indirizzi e-mail che restituiscono un mancato recapito permanente a un elenco di eliminazione a livello di account o a un elenco di eliminazione globale. Se un indirizzo e-mail non recapitabile diventa recapitabile in un secondo momento, non puoi controllarne la rimozione dall'elenco di soppressione mentre il pool di utenti è configurato per utilizzare la configurazione predefinita. Un indirizzo e-mail può rimanere nell'elenco di soppressione gestito a tempo indeterminato. AWS Per gestire gli indirizzi e-mail non recapitabili, utilizza la configurazione e-mail di Amazon SES con un elenco di eliminazione a livello di account, come descritto nella prossima sezione.

Con la configurazione e-mail predefinita, puoi utilizzare uno dei seguenti indirizzi come FROM:

- L'indirizzo e-mail di default, ovvero no-reply@verificationemail.com.

- Un indirizzo e-mail personalizzato. Prima di poter utilizzare il tuo indirizzo e-mail, devi verificarlo con Amazon SES e concedere a Amazon Cognito l'autorizzazione per utilizzarlo.

Configurazione e-mail di Amazon SES

L'applicazione potrebbe richiedere un volume di distribuzione più elevato rispetto a quello disponibile con l'opzione predefinita. Per aumentare il volume di distribuzione, usa le risorse di Amazon SES con il bacino d'utenza per inviare e-mail ai tuoi utenti. Puoi anche monitorare la tua attività di invio e-mail quando invii messaggi e-mail con la tua configurazione Amazon SES.

Prima di poter utilizzare la tua configurazione Amazon SES, devi verificare uno o più indirizzi e-mail, o un dominio, con Amazon SES. Usa un indirizzo e-mail verificato o un indirizzo da un dominio verificato come indirizzo e-mail FROM (DA) da assegnare al bacino d'utenza. Quando Amazon Cognito invia un messaggio e-mail a un utente, utilizza il tuo indirizzo e-mail chiamando automaticamente Amazon SES.

Quando utilizzi la configurazione Amazon SES, si applicano le seguenti condizioni:

- I limiti di invio dei messaggi e-mail per il bacino d'utenza sono gli stessi che valgono per l'indirizzo e-mail verificato Amazon SES nel tuo Account AWS.
- Puoi gestire i tuoi messaggi ad indirizzi e-mail non recapitabili con un elenco di eliminazione a livello di account in Amazon SES che sostituisce l'elenco di eliminazione globale. Quando utilizzi un elenco di eliminazione a livello di account, i mancati recapiti dei messaggi e-mail influiscono sulla reputazione del tuo account come mittente. Per ulteriori informazioni, consulta Utilizzo dell'elenco di eliminazione a livello di account di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Regioni di configurazione degli indirizzi e-mail di Amazon SES

Il Regione AWS luogo in cui crei un pool di utenti avrà uno dei tre requisiti per la configurazione dei messaggi e-mail con Amazon SES. Puoi inviare messaggi e-mail da Amazon SES nella stessa regione del tuo pool di utenti, in diverse regioni tra cui la stessa regione o in una o più regioni remote. Per prestazioni ottimali, invia messaggi e-mail con un'identità verificata da Amazon SES nella stessa regione del tuo pool di utenti, se ne hai l'opzione.

Categorie di requisiti regionali per le identità verificate di Amazon SES

Solo all'interno della regione

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate nello Regione AWS stesso gruppo di utenti. Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata nella stessa regione.

Compatibile con le versioni precedenti

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate nella stessa Regione AWS o in una delle seguenti regioni alternative:

- Stati Uniti orientali (Virginia settentrionale)
- US West (Oregon)
- Europa (Irlanda)

Questa funzionalità supporta la continuità per le risorse del pool di utenti che potresti aver creato per soddisfare i requisiti di Amazon Cognito al momento del lancio del servizio. I pool di utenti di quel periodo potevano inviare messaggi e-mail con identità verificate solo in un numero limitato di. Regioni AWS Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata nella stessa regione.

Regione alternativa

I tuoi pool di utenti possono inviare messaggi e-mail con identità verificate in un'alternativa Regione AWS esterna alla regione del pool di utenti. Questa configurazione si verifica quando Amazon SES non è disponibile in una regione in cui è disponibile Amazon Cognito.

La politica di autorizzazione all'invio di Amazon SES per la tua identità verificata nella regione alternativa deve considerare attendibile il responsabile del servizio Amazon Cognito della regione di origine. Per ulteriori informazioni, consulta Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita.

In alcune di queste regioni, Amazon Cognito divide i messaggi e-mail tra due regioni alternative per la configurazione e-mail predefinita di. `COGNITO_DEFAULT` In questi casi, per utilizzare un indirizzo `FROM` e-mail personalizzato, la politica di autorizzazione all'invio di Amazon SES per la tua identità verificata in ciascuna regione alternativa deve considerare attendibile il

responsabile del servizio Amazon Cognito della regione di origine. Per ulteriori informazioni, consulta Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita. Con la configurazione e-mail di Amazon SES `DEVELOPER` in queste regioni, devi utilizzare un'identità verificata nella prima regione elencata e configurarla in modo che si fidi del principale servizio Amazon Cognito nella regione del pool di utenti. Ad esempio, in un pool di utenti in Medio Oriente (Emirati Arabi Uniti), configura un'identità verificata in Europa (Francoforte) da considerare attendibile. `cognito-idp.me-central-1.amazonaws.com` Nella configurazione e-mail predefinita senza un indirizzo `FROM` e-mail personalizzato, Amazon Cognito utilizza un'identità `no-reply@verificationemail.com` verificata in ogni regione.

> **Note**
>
> Nella seguente combinazione di condizioni, devi specificare il `SourceArn` parametro EmailConfigurationcon un carattere jolly nell'elemento Regione, nel formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}` Ciò consente al tuo pool di utenti di inviare messaggi e-mail con identità verificate identiche in entrambi i tuoi Account AWS . Regioni AWS
>
> - Il tuo EmailSendingAccount è. `COGNITO_DEFAULT`
> - Vuoi usare un `FROM` indirizzo personalizzato.
> - Il tuo pool di utenti invia e-mail in una regione alternativa.
> - Il tuo pool di utenti ha una seconda regione [1]alternativa specificata nella tabella delle regioni supportate da Amazon SES che segue.

Se crei un pool di utenti in modo programmatico, con un SDK AWS , l'API o la CLI di Amazon Cognito, oppure AWS CDK AWS CloudFormation, il tuo pool di utenti invia messaggi e-mail con l'identità Amazon `SourceArn` SES specificata dal parametro per il tuo pool di utenti. EmailConfiguration L'identità Amazon SES deve occupare un indirizzo supportato Regione AWS. Se il tuo `EmailSendingAccount` è `COGNITO_DEFAULT` e non specific un parametro`SourceArn`, Amazon Cognito invia messaggi e-mail da `no-reply@verificationemail.com` utilizzando risorse nella regione in cui hai creato il bacino d'utenza.

La tabella seguente mostra Regioni AWS dove è possibile utilizzare le identità Amazon SES con Amazon Cognito.

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
Stati Uniti orientali (Virginia settentrionale)	Compatibile con le versioni precedenti	Stati Uniti occidentali (Oregon), Stati Uniti orientali (Virginia settentrionale), Europa (Irlanda).
Stati Uniti orientali (Ohio)	Compatibile con le versioni precedenti	Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Stati Uniti occidentali (California settentrionale)	Solo all'interno della regione	Stati Uniti occidentali (California settentrionale)
US West (Oregon)	Compatibile con le versioni precedenti	Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Canada (Centrale)	Compatibile con le versioni precedenti	Canada (Centrale), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Asia Pacifico (Tokyo)	Compatibile con le versioni precedenti	Asia Pacifico (Tokyo), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Seoul)	Compatibile con le versioni precedenti	Asia Pacifico (Seoul), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Mumbai)	Compatibile con le versioni precedenti	Asia Pacifico (Mumbai), Stati Uniti orientali (Virginia settentri

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
		onale), Stati Uniti occidentali (Oregon)
Asia Pacific (Hyderabad)	Regione alternativa	Asia Pacifico (Mumbai), Asia Pacifico (Singapore) [1]
Asia Pacifico (Singapore)	Compatibile con le versioni precedenti	Asia Pacifico (Singapore), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Sydney)	Compatibile con le versioni precedenti	Asia Pacifico (Sydney), Stati Uniti orientali (Virginia settentri onale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Asia Pacifico (Osaka-Locale)	Solo all'interno della regione	Asia Pacifico (Osaka-Locale)
Asia Pacifico (Giacarta)	Solo all'interno della regione	Asia Pacifico (Giacarta)
Asia Pacifico (Melbourne)	Regione alternativa	Asia Pacifico (Sydney), Asia Pacifico (Singapore) [1]
Europa (Irlanda)	Compatibile con le versioni precedenti	Stati Uniti occidentali (Oregon), Stati Uniti orientali (Virginia settentrionale), Europa (Irlanda).
Europa (Londra)	Compatibile con le versioni precedenti	Europa (Londra), Stati Uniti orientali (Virginia settentri onale), Stati Uniti occidentali (Oregon), Europa (Irlanda).
Europa (Parigi)	Solo all'interno della regione	Europa (Parigi)

Regione del pool di utenti	Opzione Regione	Regioni supportate da Amazon SES
Europa (Francoforte)	Compatibile con le versioni precedenti	Europa (Francoforte), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Europa (Irlanda)
Europa (Zurigo)	Regione alternativa	Europa (Francoforte), Europa (Londra) [1]
Europa (Stoccolma)	Solo all'interno della regione	Europa (Stoccolma)
Europa (Milano)	Solo all'interno della regione	Europa (Milano)
Europa (Spagna)	Regione alternativa	Europa (Parigi), Europa (Stoccolma) [1]
Medio Oriente (Bahrein)	Solo all'interno della regione	Medio Oriente (Bahrein)
Medio Oriente (Emirati Arabi Uniti)	Regione alternativa	Europa (Francoforte), Europa (Londra) [1]
Sud America (San Paolo)	Solo all'interno della regione	Sud America (San Paolo)
Israele (Tel Aviv)	Solo all'interno della regione	Israele (Tel Aviv)
Africa (Città del Capo)	Solo all'interno della regione	Africa (Città del Capo)

[1] Utilizzato nei pool di utenti con la configurazione e-mail predefinita. Amazon Cognito distribuisce messaggi e-mail tra identità verificate con lo stesso indirizzo e-mail in ogni regione. Per utilizzare un `FROM` indirizzo personalizzato, configuralo `EmailConfiguration` con un `SourceArn` parametro nel formato. `arn:${Partition}:ses:*:${Account}:identity/${IdentityName}`

Configurazione delle e-mail per il bacino d'utenza

Completa le fasi seguenti per configurare le impostazioni e-mail per il bacino d'utenza. A seconda delle impostazioni che usi, potresti aver bisogno di autorizzazioni IAM in Amazon SES, AWS Identity and Access Management (IAM) e Amazon Cognito.

> **Note**
>
> Le risorse da te create in questi passaggi non possono essere condivise tra Account AWS. Ad esempio, non puoi configurare un bacino d'utenza in un account e poi usarlo con un indirizzo e-mail Amazon SES in un account diverso. Se utilizzi Amazon Cognito in più di un account, ripeti questi passaggi per ognuno di essi.

Fase 1: Verifica dell'indirizzo e-mail o del dominio con Amazon SES

Prima di configurare il bacino d'utenza, se desideri eseguire una delle operazioni seguenti devi verificare uno o più indirizzi e-mail con Amazon SES:

- Utilizzare l'indirizzo e-mail personale come indirizzo FROM
- Utilizzo della configurazione di Amazon SES per gestire la distribuzione delle e-mail

Verificando l'indirizzo e-mail, confermi di esserne il proprietario, il che impedisce l'utilizzo non autorizzato.

Per ulteriori informazioni sulla verifica di un indirizzo e-mail con Amazon SES, consulta Verifica di un indirizzo e-mail nella Guida per gli sviluppatori di Amazon Simple Email Service. Per ulteriori informazioni sulla verifica di un dominio con Amazon SES,, consulta la sezione Verifying domains (verifica dei domini).

Fase 2: Spostamento dell'account all'esterno della sandbox di Amazon SES

Ometti questo passaggio se utilizzi la configurazione e-mail predefinita di Amazon Cognito.

Quando usi Amazon SES per la prima volta in qualsiasi regione Regione AWS, ti colloca Account AWS nella sandbox Amazon SES per quella regione. Amazon SES utilizza la sandbox per prevenire frodi e usi illeciti. Se utilizzi la configurazione di Amazon SES per gestire la consegna delle e-mail, perché Amazon Cognito possa inviare e-mail ai tuoi utenti devi spostare il tuo Account AWS all'esterno della sandbox.

Nella sandbox, Amazon SES impone delle limitazioni al numero di e-mail che si possono inviare e agli indirizzi raggiungibili. Puoi inviare e-mail solo a indirizzi e domini verificati con Amazon SES oppure agli indirizzi del simulatore di mailbox di Amazon SES. Mentre Account AWS resti nella sandbox, non utilizzare la configurazione di Amazon SES per applicazioni in produzione. In questo caso, Amazon Cognito non può inviare messaggi agli indirizzi e-mail degli utenti.

Per rimuovere il tuo Account AWS dalla sandbox, consulta Uscire dalla sandbox di Amazon SES nella Amazon Simple Email Service Developer Guide.

Fase 3: Concessione delle autorizzazioni e-mail ad Amazon Cognito

Prima che Amazon Cognito possa inviare e-mail ai tuoi utenti, è possibile che tu debba concedergli autorizzazioni specifiche. Le autorizzazioni concesse e il processo che utilizzi per concederle dipendono dal fatto che tu stia utilizzando la configurazione e-mail predefinita o la configurazione Amazon SES.

Per concedere le autorizzazioni per utilizzare la configurazione e-mail predefinita

Completa questo passaggio solo se configuri il tuo pool di utenti su Invia email con Cognito o se hai impostato su`EmailSendingAccount`. `COGNITO_DEFAULT`

Con la configurazione e-mail predefinita, il tuo pool di utenti può inviare messaggi e-mail con uno dei seguenti indirizzi.

- L'indirizzo predefinitono-`reply@verificationemail.com`.
- Un indirizzo FROM personalizzato dai tuoi indirizzi e-mail o domini verificati in Amazon SES.

Se utilizzi un indirizzo personalizzato, Amazon Cognito ha bisogno di ulteriori autorizzazioni per inviare e-mail agli utenti da questo indirizzo. Queste autorizzazioni sono concesse da una politica di autorizzazione all'invio per l'indirizzo o il dominio in Amazon SES. Se si utilizza la console Amazon Cognito per aggiungere un indirizzo personalizzato al bacino d'utenza, la policy viene automaticamente collegata all'indirizzo e-mail verificato di Amazon SES. Tuttavia, se configuri il tuo pool di utenti al di fuori della console, ad esempio utilizzando l' AWS CLI API Amazon Cognito, devi allegare la policy utilizzando la console Amazon SES o l'PutIdentityPolicyAPI.

> **Note**
>
> È possibile configurare un indirizzo FROM solo in un dominio verificato utilizzando la AWS CLI o l'API Amazon Cognito.

Una policy di autorizzazione all'invio consente o nega l'accesso in base alle risorse dell'account che utilizzano Amazon Cognito per richiamare Amazon SES. Per ulteriori informazioni sulle policy basate su risorse, consulta Guida per l'utente di IAM. Puoi anche trovare esempi di policy basate su risorse nella Guida per gli sviluppatori di Amazon SES.

Example Policy di autorizzazione di invio

L'esempio seguente di policy di autorizzazione di invio garantisce ad Amazon Cognito la possibilità limitata di utilizzare un'identità verificata di Amazon SES. Amazon Cognito può inviare e-mail solo quando lo fa per conto sia del bacino d'utenza nella condizione `aws:SourceArn` sia dell'account nella condizione `aws:SourceAccount`.

Regions with Amazon SES

La tua politica di autorizzazione all'invio nella regione del pool di utenti o nella regione alternativa deve consentire al responsabile del servizio Amazon Cognito di inviare messaggi e-mail. Per ulteriori informazioni, consulta la tabella Regioni. Se la regione del pool di utenti corrisponde ad almeno un valore nella regione Amazon SES, configura la politica di autorizzazione all'invio con il servizio principale globale nell'esempio seguente.

```
{
    "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 non è disponibile in tutti gli opt-in in Regioni AWS cui è disponibile Amazon Cognito. Il Medio Oriente (Emirati Arabi Uniti) ne è un esempio e può inviare e-mail con identità verificate solo in Europa (Francoforte) (). `eu-central-1` Nei pool di utenti con la configurazione e-mail predefinita, Amazon Cognito invia anche messaggi e-mail con un'identità verificata in ciascuna delle due regioni. Nel caso del Medio Oriente (Emirati Arabi Uniti), la regione aggiuntiva è l'Europa (Londra). È necessario aggiornare la politica di autorizzazione all'invio in entrambe le regioni.

La tua politica di autorizzazione all'invio in ciascuna delle regioni alternative deve consentire al responsabile del servizio Amazon Cognito nella regione di attivazione del pool di utenti di inviare messaggi e-mail. Per ulteriori informazioni, consulta la tabella delle regioni. Se la tua regione è contrassegnata come Regione alternativa, configura le politiche di autorizzazione all'invio con il responsabile del servizio regionale, come nell'esempio seguente. Sostituisci l'identificatore regionale di esempio *me-central-1* con l'ID regionale richiesto, se necessario.

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

Per ulteriori informazioni sulla sintassi della policy, consulta la sezione Policy di autorizzazione di invio di Amazon SES nella Guida per gli sviluppatori di Amazon Simple Email Service.

Per altri esempi, consulta la sezione Amazon SES sending authorization policy examples (Esempi di policy di autorizzazione di invio di Amazon SES) nella Guida per gli sviluppatori di Amazon Simple Email Service.

Come concedere le autorizzazioni per utilizzare la configurazione Amazon SES

Se configuri il bacino d'utenza in modo da utilizzare la configurazione di Amazon SES, Amazon Cognito avrò bisogno di ulteriori autorizzazioni per richiamare Amazon SES per tuo conto quando invia e-mail agli utenti. Questa autorizzazione è concessa con il servizio IAM.

Quando configuri il bacino d'utenza con questa opzione, Amazon Cognito crea un ruolo collegato al servizio, che è un tipo di ruolo IAM, nel tuo Account AWS. Questo ruolo contiene le autorizzazioni che permettono ad Amazon Cognito di accedere ad Amazon SES e di inviare messaggi e-mail con il tuo indirizzo.

Amazon Cognito crea il tuo ruolo collegato al servizio con AWS le credenziali della sessione utente che imposta la configurazione. Le autorizzazioni IAM di questa sessione devono includere l'azione `iam:CreateServiceLinkedRole`. Per ulteriori informazioni sulle autorizzazioni in IAM, consulta Gestione degli accessi alle AWS risorse nella IAM User Guide.

Per ulteriori informazioni sul ruolo collegato al servizio creato da Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Fase 4: configurazione del bacino d'utenza

Completa la procedura seguente se vuoi configurare il bacino d'utenza con uno dei seguenti elementi:

- Un indirizzo FROM personalizzato che compare come mittente dell'e-mail

- Un indirizzo REPLY-TO personalizzato che riceve i messaggi inviati dagli utenti all'indirizzo FROM
- La tua configurazione di Amazon SES

> **Note**
>
> Se la tua identità verificata è un indirizzo e-mail, Amazon Cognito imposta tale indirizzo e-mail come indirizzo e-mail FROM e REPLY-TO per impostazione predefinita. Tuttavia, se la tua identità verificata è un dominio, devi fornire un valore per gli indirizzi e-mail FROM e REPLY-TO. Ad esempio, se il tuo dominio verificato è example.com, puoi impostare no-reply@example.com sia come indirizzo email FROM che REPLY-TO.

Ometti questa procedura se desideri utilizzare la configurazione e l'indirizzo e-mail predefiniti di Amazon Cognito.

Per configurare il bacino d'utenza per utilizzare un indirizzo e-mail personalizzato

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali. AWS
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Messaging (Messaggistica), individua la voce Email configuration (configurazione e-mail), e scegli Edit (Modifica).
5. Nella pagina Edit email configuration (Modifica della configurazione e-mail) seleziona l'opzione Send email from Amazon SES (Invia e-mail da Amazon SES) o Send email with Amazon Cognito (Invia e-mail con Amazon Cognito). Puoi personalizzare la Regione SES, il Set di configurazione e il nome del mittente FROM (DA) solo selezionando l'opzione Send email from Amazon SES (Invia e-mail da Amazon SES).
6. Per utilizzare un indirizzo FROM (DA) personalizzato, completa la seguente procedura:
   a. In SES Region (Regione SES), scegli la regione che contiene l'indirizzo e-mail verificato.
   b. In FROM email address (Indirizzo e-mail FROM, (DA)), selezionare l'indirizzo e-mail. Usa un indirizzo e-mail che hai verificato con Amazon SES.
   c. (Facoltativo) Alla voce Configuration set (Set di configurazione), scegli un set di configurazione da utilizzare con Amazon SES. La creazione e il salvataggio di questa modifica genera un ruolo collegato al servizio.

d. (Facoltativo) Alla voce Indirizzo del mittente FROM (DA), inserisci un indirizzo e-mail. Puoi fornire solo il tuo indirizzo e-mail o il tuo indirizzo e-mail e un nome descrittivo nel formato `Jane Doe <janedoe@example.com>`.

e. (Facoltativo) Alla voce REPLY-TO email address (RISPONDI A indirizzo e-mail), inserire l'indirizzo e-mail in cui si desidera ricevere i messaggi inviati dagli utenti all'indirizzo FROM (DA).

7. Scegli Save changes (Salva modifiche).

Argomenti correlati

- Personalizzazione dei messaggi di verifica delle e-mail
- Personalizzazione dei messaggi di invito degli utenti

Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito

Alcuni eventi di Amazon Cognito per il tuo bacino d'utenza potrebbero causare l'invio di SMS da parte di Amazon Cognito ai tuoi utenti. Ad esempio, se configuri il bacino d'utenza in modo da richiedere la verifica tramite telefono, Amazon Cognito invia un SMS quando un utente registra un nuovo account nell'app o quando reimposta la password. A seconda dell'azione che attiva il messaggio SMS, il messaggio può contenere un codice di verifica, una password temporanea o un messaggio di benvenuto.

Amazon Cognito utilizza Amazon Simple Notification Service (SNS) per l'invio di SMS. Se questa è la prima volta che invii un SMS tramite Amazon Cognito o Amazon SNS, sarai inserito in un ambiente sandbox in Amazon SNS. Nell'ambiente sandbox è possibile testare le applicazioni per gli SMS. Nella sandbox, i messaggi possono essere inviati solo a numeri di telefono verificati.

Amazon SNS addebita il costo dei messaggi SMS. Per ulteriori informazioni, consulta Prezzi di Amazon SNS.

> **Note**
>
> A causa del volume di traffico SMS indesiderato in tutto il mondo, alcuni governi impongono barriere tra mittenti e destinatari dei messaggi SMS. Quando utilizzi i messaggi SMS per l'autenticazione MFA e gli aggiornamenti utente, devi adottare misure aggiuntive per

assicurarti che i messaggi vengano recapitati. Devi inoltre monitorare le normative relative ai messaggi SMS nei paesi in cui potrebbero risiedere gli utenti e mantenere aggiornata la configurazione dei messaggi SMS. Per ulteriori informazioni, consultare Messaggi di testo mobili (SMS) nella Guida per gli sviluppatori di Amazon Simple Notification Service.
L'uso di messaggi SMS per autenticare e verificare gli utenti non è una best practice di sicurezza. I numeri di telefono possono cambiare proprietario e potrebbero non rappresentare in modo affidabile un fattore di autenticazione MFA elementi che possiedi per gli utenti. Implementa invece l'autenticazione MFA TOTP nell'app o con il gestore dell'identità digitale (IdP) di terze parti. Puoi anche creare fattori di autenticazione personalizzati aggiuntivi con Trigger Lambda di richieste di autenticazione personalizzate.

Amazon Cognito invia messaggi SMS agli utenti con un codice che possono inserire. Nella tabella seguente vengono mostrati gli eventi che possono generare un messaggio SMS.

Opzioni relative ai messaggi

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Password dimenticata	ForgotPassword	Posta elettronica, SMS	code	No	N/D
Invito	AdminCreateUser	Posta elettronica, SMS	code	Sì	Messaggio di invito
Autoregistrazione	SignUp	E-mail, SMS	codice, link	Sì	messaggio di verifica
Verifica dell'indirizzo e-mail o del numero di telefono	UpdateUserAttributes	E-mail, SMS	code	Sì	Messaggio di verifica

Attività	Operazione API	Opzioni di distribuzione	Opzioni di formato	Personalizzabile	Modello di messaggio
Autenticazione a più fattori (MFA)	AdminInitiateAuth, InitiateAuth	SMS, app di autenticazione	code	Sì[1]	Messaggio MFA

[1] Per messaggi SMS.

Prima impostazione degli SMS nei bacini d'utenza di Amazon Cognito

Amazon Cognito utilizza Amazon SNS per inviare messaggi SMS ai tuoi bacini d'utenza. Puoi utilizzare un Trigger Lambda del mittente di SMS personalizzato, se desideri servirti delle tue risorse per inviare messaggi SMS. La prima volta che configuri Amazon SNS per inviare messaggi di testo SMS in particolare Regione AWS, Amazon SNS ti inserisce Account AWS nella sandbox SMS per quella regione. Amazon SNS utilizza la sandbox per prevenire frodi e abusi e per soddisfare i requisiti di conformità. Quando ti Account AWS trovi nella sandbox, Amazon SNS impone alcune restrizioni. Ad esempio, puoi inviare SMS a un massimo di 10 numeri di telefono verificati con Amazon SNS. Mentre Account AWS resti nella sandbox, non utilizzare la configurazione di Amazon SNS per applicazioni in produzione. Quando sei nella sandbox, Amazon Cognito non può inviare messaggi ai numeri di telefono degli utenti.

Per inviare SMS agli utenti del bacino d'utenza

1. Preparazione di un ruolo IAM che Amazon Cognito può utilizzare per inviare messaggi SMS con Amazon SNS
2. Scegli l'opzione Regione AWS per i messaggi SMS di Amazon SNS
3. Ottenere un'identità di origine per l'invio di messaggi SMS a numeri di telefono USA
4. Verifica di trovarti nella sandbox SMS
5. Sposta il tuo account all'esterno della sandbox di Amazon SNS
6. Verifica i numeri di telefono per Amazon Cognito in Amazon SNS
7. Completamento della configurazione del pool di utenti in Amazon Cognito

Preparazione di un ruolo IAM che Amazon Cognito può utilizzare per inviare messaggi SMS con Amazon SNS

Quando invii un SMS dal tuo pool di utenti, Amazon Cognito assume un ruolo IAM nel tuo account. Amazon Cognito utilizza l'autorizzazione `sns:Publish` assegnata a quel ruolo per inviare SMS agli utenti. Nella console Amazon Cognito, è possibile impostare un valore nel campo IAM role selection (Selezione del ruolo IAM) della scheda Messaging (Messaggistica) del pool di utenti, nell'area SMS, oppure effettuare questa selezione durante la procedura guidata di creazione del pool di utenti.

Il seguente esempio di policy di attendibilità dei ruoli IAM garantisce al pool di utenti di Amazon Cognito una capacità limitata di assumere un ruolo IAM. Amazon Cognito può assumere il ruolo solo quando lo fa per conto dei pool di identità nella condizione `aws:SourceArn` e dell' Account AWS nella condizione `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>"
            }
        }
    }]
}
```

Puoi specificare un valore esatto per l'ARN del pool di utenti o un ARN con caratteri jolly nel valore della condizione `aws:SourceArn`. Cerca gli ARN dei tuoi pool di utenti in AWS Management Console o con una DescribeUserPoolrichiesta API.

Per ulteriori informazioni sui ruoli IAM e le policy di attendibilità, consulta Termini e concetti dei ruoli nella Guida per l'utente di AWS Identity and Access Management .

Scegli l'opzione Regione AWS per i messaggi SMS di Amazon SNS

In alcuni Regioni AWS, puoi scegliere la regione che contiene le risorse Amazon SNS che desideri utilizzare per i messaggi SMS di Amazon Cognito. Regione AWS Ovunque sia disponibile Amazon Cognito, ad eccezione dell'Asia Pacifico (Seoul), puoi utilizzare le risorse Amazon SNS nel luogo in Regione AWS cui hai creato il tuo pool di utenti. Per rendere l'invio di messaggi SMS più veloce e affidabile quando puoi scegliere tra più regioni, usa le risorse Amazon SNS nella stessa regione del tuo bacino d'utenza.

> **Note**
>
> In AWS Management Console, puoi modificare la regione per le risorse SMS solo dopo essere passato alla nuova esperienza della console Amazon Cognito.

Scegli una regione per le risorse SMS nel passaggio Configure message delivery (Configura invio di messaggi) della procedura guidata per la configurazione del nuovo bacino d'utenza. È possibile anche scegliere Edit (Modifica) sotto SMS nella schedaMessaging (Messaggistica) di un bacino d'utenza esistente.

Al momento del lancio, per alcuni Regioni AWS, Amazon Cognito inviava messaggi SMS con risorse Amazon SNS in una regione alternativa. Per impostare la tua regione preferita, usa il `SnsRegion` parametro dell'SmsConfigurationTypeoggetto per il tuo pool di utenti. Quando crei a livello di programmazione una risorsa del bacino d'utenza di Amazon Cognito in una Regione di Amazon Cognito inclusa nella tabella seguente e non fornisci un parametro `SnsRegion`, il bacino d'utenza può inviare messaggi SMS utilizzando le risorse Amazon SNS di una Regione di Amazon SNS legacy.

I pool di utenti di Amazon Cognito in Asia Pacifico (Seoul) Regione AWS devono utilizzare la configurazione Amazon SNS nella regione Asia Pacifico (Tokyo).

La quota di spesa di Amazon SNS è impostata su 1,00 USD al mese per tutti i nuovi account. Potresti aver aumentato il limite di spesa in un account Regione AWS che utilizzi con Amazon Cognito. Prima di modificare Regione AWS i messaggi SMS di Amazon SNS, apri una richiesta di aumento della quota nel AWS Support Center per aumentare il limite nella nuova regione. Per ulteriori informazioni, consulta Richiesta di aumento della quota di spesa mensile per l'invio di SMS per Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Puoi inviare messaggi SMS per qualsiasi Regione di Amazon Cognito inclusa nella tabella seguente utilizzando le risorse Amazon SNS della Regione di Amazon SNS corrispondente.

Regione di Amazon Cognito	Regione di Amazon SNS
Stati Uniti orientali (Ohio)	Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale)
Canada (Centrale)	Canada (Centrale), Stati Uniti orientali (Virginia settentrionale)
Europa (Francoforte)	UE (Francoforte), UE (Irlanda)
Europa (Londra)	Europa (Londra), Europa (Irlanda)
Asia Pacifico (Seoul)	Asia Pacifico (Tokyo)
Stati Uniti orientali (Virginia settentrionale)	Stati Uniti orientali (Virginia settentrionale)
Stati Uniti occidentali (California settentrionale)	Stati Uniti occidentali (California settentrionale)
US West (Oregon)	US West (Oregon)
Asia Pacifico (Mumbai)	Asia Pacifico (Mumbai), Asia Pacifico (Singapore)
Asia Pacific (Hyderabad)	Asia Pacific (Hyderabad)
Asia Pacifico (Singapore)	Asia Pacifico (Singapore)
Asia Pacifico (Sydney)	Asia Pacifico (Sydney)
Asia Pacifico (Tokyo)	Asia Pacifico (Tokyo)
Asia Pacifico (Giacarta)	Asia Pacifico (Giacarta)
Asia Pacifico (Osaka-Locale)	Asia Pacifico (Osaka-Locale)
Asia Pacifico (Melbourne)	Asia Pacifico (Melbourne)
Europa (Irlanda)	Europa (Irlanda)

Regione di Amazon Cognito	Regione di Amazon SNS
Europa (Parigi)	Europa (Parigi)
Europa (Stoccolma)	Europa (Stoccolma)
Europa (Milano)	Europa (Milano)
Europa (Spagna)	Europa (Spagna)
Medio Oriente (Bahrein)	Medio Oriente (Bahrein)
Sud America (San Paolo)	Sud America (San Paolo)
Israele (Tel Aviv)	Israele (Tel Aviv)
Africa (Città del Capo)	Africa (Città del Capo)
Medio Oriente (Emirati Arabi Uniti)	Medio Oriente (Emirati Arabi Uniti)
Europa (Zurigo)	Europa (Zurigo)

Ottenere un'identità di origine per l'invio di messaggi SMS a numeri di telefono USA

Se intendi inviare messaggi SMS a numeri di telefono negli Stati Uniti, devi ottenere un'identità di origine, indipendentemente dalla creazione di un ambiente di test sandbox SMS o di produzione.

A partire dal 1° giugno 2021, gli operatori statunitensi richiedono agli utenti di fornire un'identità di origine per inviare messaggi a numeri di telefono degli Stati Uniti. Se non disponi ancora di un'identità di origine, devi richiederne una. Per informazioni su come ottenere un'identità di origine, consulta Richiesta di un numero nella Guida per l'utente di Amazon Pinpoint.

Se operi nei seguenti settori Regioni AWS, devi aprire un AWS Support ticket per ottenere un'identità di origine. Per le istruzioni, consulta Richiesta di supporto per la messaggistica SMS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

- Stati Uniti orientali (Ohio)
- Europa (Stoccolma)
- Europa (Parigi)

- Europa (Milano)
- Middle East (Bahrain)
- Sud America (San Paolo)
- Stati Uniti occidentali (California settentrionale)

Quando hai più di un'identità di origine nella stessa identità Regione AWS, Amazon SNS sceglie un tipo di identità di origine nel seguente ordine di priorità: codice breve, 10DLC, numero verde. Non puoi modificare questa priorità. Per ulteriori dettagli, consulta Domande frequenti su Amazon SNS.

Verifica di trovarti nella sandbox SMS

Utilizza la procedura seguente per confermare se ti trovi nella sandbox SMS. Ripeti l'operazione per ogni Regione AWS area in cui hai pool di utenti Amazon Cognito di produzione.

Verifica dello stato della sandbox SMS nella console Amazon Cognito

Per confermare di trovarti nella sandbox SMS

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS .
2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco.
4. Scegli la scheda Messaging (Messaggistica).
5. Nella sezione Configurazione SMS, espandi Move to Amazon SNS production environment (passare all'ambiente di produzione Amazon SNS). Se l'account si trova nella sandbox SMS, in Amazon Cognito verrà visualizzato il messaggio seguente:

   ```
   You are currently in the SMS Sandbox and cannot send SMS messages to unverified numbers.
   ```

   Se il messaggio non compare, significa che qualcuno ha già eseguito la configurazione dei messaggi SMS nel tuo account. Passa a Completamento della configurazione del pool di utenti in Amazon Cognito.
6. Seleziona il link Amazon SNS nel messaggio. La console SNS viene aperta in una nuova scheda.
7. Verifica di trovarti nell'ambiente sandbox. Il messaggio della console indica lo stato della sandbox e Regione AWS, come segue:

```
This account is in the SMS sandbox in US East (N. Virginia).
```

Sposta il tuo account all'esterno della sandbox di Amazon SNS

Se stai eseguendo il test dell'app e devi solo inviare messaggi SMS ai numeri di telefono che gli amministratori possono verificare, ignora questo passaggio.

Per utilizzare la tua app in produzione, sposta il tuo account dall'ambiente della sandbox SMS a quello della produzione. Dopo aver configurato un'identità di origine Regione AWS che contiene le risorse Amazon SNS che desideri vengano utilizzate da Amazon Cognito, puoi verificare i numeri di telefono degli Stati Uniti mentre i Account AWS tuoi rimangono nella sandbox SMS. Quando il tuo ambiente Amazon SNS è in produzione, non devi verificare i numeri di telefono utente in Amazon SNS per inviare messaggi SMS agli utenti.

Per le istruzioni dettagliate, consulta Uscita dalla sandbox SMS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Verifica i numeri di telefono per Amazon Cognito in Amazon SNS

Se hai spostato il tuo account al di fuori dell'ambiente di sperimentazione (sandbox) SMS, ignora questo passaggio.

Quando sei nella sandbox SMS, puoi inviare messaggi a qualsiasi numero di telefono verificato con Amazon SNS.

Per verificare un numero di telefono, procedi come segue:

1. Aggiungi un Numero di telefono di destinazione sandbox nella sezione Text messaging (SMS) della console Amazon SNS.
2. Ricevi un SMS con un codice al numero di telefono fornito.
3. Inserisci il Codice di verifica ricevuto via SMS nella console Amazon SNS.

Per le istruzioni dettagliate, consulta Aggiunta e verifica dei numeri di telefono nella sandbox SMSnella Guida per gli sviluppatori di Amazon Simple Notification Service.

> **Note**
>
> Amazon SNS limita il numero di numeri di telefono di destinazione che puoi verificare mentre sei nella sandbox SMS. Per maggiori dettagli, consulta Sandbox SMS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Completamento della configurazione del pool di utenti in Amazon Cognito

Torna alla scheda del browser in cui stavi creando o modificando il bacino d'utenza. Completare la procedura. Dopo aver aggiunto correttamente la configurazione dei messaggi SMS al pool di utenti, Amazon Cognito invia un messaggio di prova a un numero di telefono interno per verificare che la configurazione funzioni. Amazon SNS addebita il costo di ogni messaggio SMS di prova.

Utilizzo di token con bacini d'utenza

Autentica gli utenti e concedi l'accesso alle risorse con i token. I token includono le attestazioni che sono informazioni sull'utente. Il token ID contiene le attestazioni relative all'identità, come il nome, il cognome e l'indirizzo e-mail dell'utente. Il token di accesso contiene le attestazioni, ad esempio `scope`, che l'utente autenticato può utilizzare per accedere alle API di terze parti, alle operazioni API self-service degli utenti di Amazon Cognito e all'Endpoint UserInfo. I token ID e di accesso includono entrambi un'attestazione `cognito:groups` che contiene l'appartenenza al gruppo dell'utente nel pool di utenti. Per ulteriori informazioni sui pool di utenti, consulta Aggiunta di gruppi a un bacino d'utenza.

Amazon Cognito dispone anche di token di aggiornamento che puoi usare per ottenere nuovi token o revocare i token esistenti. Aggiorna un token per recuperare nuovi ID e token di accesso. Revoca un token per revocare l'accesso utente consentito dai token di aggiornamento.

Amazon Cognito emette token come stringhe con codifica Base64. Puoi decodificare qualsiasi ID Amazon Cognito o token di accesso da base64 a JSON in chiaro. I token di aggiornamento di Amazon Cognito sono crittografati, opachi agli utenti e agli amministratori dei pool di utenti e possono essere letti solo dal pool di utenti.

Autenticazione con i token

Quando un utente accede alla tua applicazione, Amazon Cognito verifica le informazioni di accesso. Se l'accesso avviene correttamente, Amazon Cognito crea una sessione e restituisce un token ID,

un token di accesso e un token di aggiornamento per l'utente autenticato. Puoi utilizzare i token per concedere agli utenti l'accesso a risorse a valle e API come Gateway Amazon API. Oppure, puoi scambiarli con credenziali AWS temporanee per accedere ad altri Servizi AWS.

Archiviazione dei token

La tua app deve essere in grado di memorizzare token di varie dimensioni. La dimensione del token può cambiare per motivi tra cui, a titolo esemplificativo, altre attestazioni, modifiche negli algoritmi di codifica e modifiche negli algoritmi di crittografia. Quando abiliti la revoca dei token nel pool di utenti, Amazon Cognito aggiunge ulteriori richieste ai token web JSON, aumentandone le dimensioni. Le nuove attestazioni `origin_jti` e `jti` vengono aggiunte ai token di accesso e ID. Per ulteriori informazioni sulla revoca dei token, consulta Revoca dei token.

> **Important**
>
> Come best practice, proteggere tutti i token durante il transito e lo storage nel contesto dell'applicazione. I token possono contenere informazioni identificative personali sugli utenti e informazioni sul modello di sicurezza utilizzato per bacino d'utenza.

Personalizzazione dei token

Puoi personalizzare i token di accesso e ID trasferiti da Amazon Cognito alla tua app. In un Trigger Lambda di pre-generazione del token, puoi aggiungere, modificare e sopprimere le richieste relative ai token. Il trigger di pre-generazione di token è una funzione Lambda a cui Amazon Cognito invia un set predefinito di richieste. Le richieste includono ambiti OAuth 2.0, appartenenza a gruppi di pool di utenti, attributi utente e altro. La funzione può quindi cogliere l'occasione per apportare modifiche al runtime e restituire richieste di token aggiornate ad Amazon Cognito.

Costi aggiuntivi vengono applicati alla personalizzazione dei token di accesso con gli eventi della versione 2. Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Argomenti

- Utilizzo di token ID

- Utilizzo del token di accesso
- Utilizzo del token di aggiornamento
- Revoca dei token
- Verifica di un JSON Web Token
- Caching dei token

Utilizzo di token ID

Il token ID è un token web JSON che contiene richieste relative all'identità dell'utente autenticato, ad esempio `name`, `email` e `phone_number`. Puoi utilizzare queste informazioni sull'identità all'interno dell'applicazione. Il token ID può essere utilizzato anche per autenticare utenti in relazione al server di risorse o alle applicazioni server. È inoltre possibile utilizzare un token ID al di fuori dell'applicazione con le operazioni dell'API Web. In questi casi, prima di poter considerare attendibile qualsiasi attestazione all'interno del token ID, devi verificare la firma del token ID. Consulta Verifica di un JSON Web Token.

Puoi configurare il periodo di scadenza del token ID su qualsiasi valore compreso tra 5 minuti e 1 giorno. Puoi configurare questo valore per il client dell'app.

> **Important**
>
> Quando l'utente accede con l'interfaccia utente ospitata o un provider di identità federato (IdP), Amazon Cognito imposta cookie di sessione validi per 1 ora. Se utilizzi l'interfaccia utente o la federazione ospitata e specifichi una durata minima inferiore a 1 ora per i token di accesso e ID, la sessione rimarrà valida per gli utenti comunque fino alla scadenza del cookie. Se l'utente dispone di token che scadono durante la sessione di un'ora, può aggiornare i token senza dover eseguire nuovamente l'autenticazione.

Intestazione del token ID

L'intestazione contiene due informazioni: l'ID della chiave (`kid`) e l'algoritmo (`alg`).

```
{
"kid" : "1234example=",
"alg" : "RS256"
}
```

`kid`

L'ID della chiave . Il valore indica quale chiave è utilizzata per proteggere la firma JWS (JSON Web Signature) del token. È possibile visualizzare gli ID delle chiavi di firma del pool di utenti nell'endpoint `jwks_uri`.

Per ulteriori informazioni sul `kid` parametro, consulta la sezione parametro di intestazione dell'ID della chiave (kid).

`alg`

L'algoritmo di crittografia utilizzato da Amazon Cognito per proteggere il token di accesso. I bacini d'utenza utilizzano un algoritmo crittografico RS256, ovvero una firma RSA con SHA-256.

Per ulteriori informazioni sul `alg` parametro, consultare la sezione relativa al parametro di intestazione dell'algoritmo (alg).

Payload predefinito del token ID

Questo è un esempio di payload derivante da un token ID. Contiene attestazioni relative all'utente autenticato. Per ulteriori informazioni sulle attestazioni standard OpenID Connect (OIDC), consulta l'elenco delle attestazioni standard OIDC. Puoi aggiungere rivendicazioni di tua progettazione con un. Trigger Lambda di pre-generazione del token

```
<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`

L'identificatore univoco (UUID), o soggetto, dell'utente autenticato. Il nome utente potrebbe non essere univoco nel pool di utenti. l'attestazione `sub` è il modo migliore per identificare un determinato utente.

`cognito:groups`

Una serie di nomi di gruppi del pool di utenti che includono l'utente come membro. I gruppi possono essere un identificatore da presentare all'app oppure possono generare una richiesta per un ruolo IAM preferito da un pool di identità.

`cognito:preferred_role`

L'ARN del ruolo IAM associato al gruppo dell'utente con la priorità più alta nel pool di utenti. Per ulteriori informazioni su come il pool di utenti seleziona questa attestazione ruolo, consulta Assegnazione dei valori di priorità ai gruppi.

`iss`

Il provider di identità che ha emesso il token. L'attestazione ha il seguente formato.

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

`cognito:username`

Il nome utente nel pool di utenti.

`nonce`

L'attestazione `nonce` proviene da un parametro con lo stesso nome che puoi aggiungere alle richieste al tuo endpoint OAuth 2.0 `authorize`. Quando si aggiunge il parametro, l'attestazione `nonce` è inclusa nel token ID emesso da Amazon Cognito e puoi usarla per proteggerti dagli attacchi di tipo replay. Se non fornisci un valore `nonce` nella tua richiesta, Amazon Cognito genera e convalida automaticamente un nonce quando esegui l'autenticazione tramite un provider di identità di terze parti, quindi lo aggiunge come attestazione `nonce` al token ID. L'implementazione dell'attestazione `nonce` in Amazon Cognito si basa su standard OIDC.

`origin_jti`

Un identificatore di revoca del token associato al token di aggiornamento dell'utente. Amazon Cognito fa riferimento all'`origin_jti`affermazione quando verifica se hai revocato il token dell'utente con l'operazione Endpoint Revoke o l'RevokeTokenAPI. Quando revochi un token, Amazon Cognito invalida tutti i token di accesso e ID con lo stesso valore `origin_jti`.

`cognito:roles`

Una serie di nomi di ruoli IAM associati ai gruppi dell'utente. A ogni gruppo di pool di utenti può essere associato un ruolo IAM. Questo array rappresenta tutti i ruoli IAM per i gruppi degli utenti, a prescindere dalla precedenza. Per ulteriori informazioni, consulta Aggiunta di gruppi a un bacino d'utenza.

`aud`

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `client_id` del token di accesso.

`identities`

Il contenuto dell'attributo `identities` dell'utente. L'attributo contiene informazioni su ogni profilo di gestore dell'identità digitale di terze parti collegato a un utente, tramite accesso federato o collegando un utente federato a un profilo locale. Queste informazioni contengono il nome del gestore, il relativo ID univoco e altri metadati.

`token_use`

Lo scopo previsto del token. In un token ID, il valore è `id`.

`auth_time`

L'ora di fine dell'autenticazione dell'utente, in formato Unix.

`exp`

L'ora di scadenza del token dell'utente, in formato Unix.

`iat`

L'ora di emissione del token dell'utente da parte di Amazon Cognito, in formato Unix.

`jti`

L'identificatore univoco del JWT.

Il token ID può contenere le attestazioni standard OIDC definite nelle attestazioni standard OIDC. Il token ID può contenere inoltre gli attributi personalizzati definiti nel bacino d'utenza. Amazon Cognito scrive i valori degli attributi personalizzati nel token ID come stringhe indipendentemente dal tipo di attributo.

> **Note**
>
> Gli attributi personalizzati del pool di utenti hanno sempre il prefisso. `custom:`

Firma del token ID

La firma del token ID viene calcolata in base all'intestazione e al payload del token JWT. Verifica la firma del token prima di accettare le attestazioni in qualsiasi token ID ricevuto dall'app. Per ulteriori informazioni, consulta Verifica di un JSON Web Token.

Utilizzo del token di accesso

Il token di accesso al bacino d'utenza contiene attestazioni relative all'utente autenticato, un elenco dei gruppi dell'utente e un elenco di ambiti. Lo scopo del token di accesso è autorizzare le operazioni API. Il pool di utenti accetta i token di accesso per autorizzare le operazioni self-service degli utenti. Ad esempio, puoi utilizzare il token di accesso per consentire agli utenti di accedere per aggiornare, modificare o eliminare gli attributi utente.

Con gli ambiti OAuth 2.0 in un token di accesso, determinato dagli ambiti personalizzati che aggiungi al pool di utenti, puoi autorizzare l'utente a recuperare informazioni da un'API. Gateway Amazon

API, ad esempio, supporta l'autorizzazione con token di accesso di Amazon Cognito. Puoi compilare un'autorizzazione REST API con le informazioni del pool di utenti o utilizzare Amazon Cognito come un'autorizzazione token web JSON per un'API HTTP. Per generare un token di accesso con ambiti personalizzati, è necessario effettuare la richiesta tramite gli endpoint pubblici del pool di utenti.

Il token di accesso dell'utente è l'autorizzazione a richiedere ulteriori informazioni sugli attributi dell'utente all'Endpoint UserInfo. Inoltre, è anche l'autorizzazione a leggere e scrivere gli attributi dell'utente. Il livello di accesso agli attributi autorizzato dal token di accesso dipende dalle autorizzazioni assegnate al client dell'app e dagli ambiti forniti nel token.

Il token di accesso è un token Web JSON (JWT). L'intestazione per il token di accesso ha la stessa struttura del token ID. Amazon Cognito firma i token di accesso con una chiave diversa da quella usata per la firma dei token ID. Il valore di un'attestazione ID chiave di accesso (`kid`) non corrisponde al valore dell'attestazione `kid` di un token ID nella stessa sessione utente. Verifica nel codice dell'app i token ID e di accesso in modo indipendente. Non considerare attendibili le attestazioni contenute in un token di accesso finché non verifichi la firma. Per ulteriori informazioni, consulta Verifica di un JSON Web Token. È possibile configurare il periodo di scadenza del token di accesso su qualsiasi valore compreso tra 5 minuti e 1 giorno. Puoi configurare questo valore per il client dell'app.

> **Important**
>
> Per i token ID e di accesso non specificare un valore minimo inferiore a un'ora se utilizzi l'interfaccia utente ospitata. Amazon Cognito HostedUI utilizza cookie validi per un'ora. Inserendo un minimo di meno di un'ora, non si otterrà un tempo di scadenza inferiore.

Intestazione del token di accesso

L'intestazione contiene due informazioni: l'ID della chiave (`kid`) e l'algoritmo (`alg`).

```
{
"kid" : "1234example="
"alg" : "RS256",
}
```

`kid`

L'ID della chiave . Il valore indica quale chiave è utilizzata per proteggere la firma JWS (JSON Web Signature) del token. È possibile visualizzare gli ID delle chiavi di firma del pool di utenti nell'endpoint `jwks_uri`.

Per ulteriori informazioni sul `kid` parametro, consulta la sezione parametro di intestazione dell'ID della chiave (kid).

`alg`

L'algoritmo di crittografia utilizzato da Amazon Cognito per proteggere il token di accesso. I bacini d'utenza utilizzano un algoritmo crittografico RS256, ovvero una firma RSA con SHA-256.

Per ulteriori informazioni sul `alg` parametro, consultare la sezione relativa al parametro di intestazione dell'algoritmo (alg).

Payload predefinito del token di accesso

Questo è un esempio di payload di un token di accesso. Per ulteriori informazioni, consultare la sezione relativa alle attestazioni JWT. Puoi aggiungere rivendicazioni personalizzate con unTrigger Lambda di pre-generazione del token.

```
<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

L'identificatore univoco (UUID), o soggetto, dell'utente autenticato. Il nome utente potrebbe non essere univoco nel pool di utenti. l'attestazione `sub` è il modo migliore per identificare un determinato utente.

cognito:groups

Una serie di nomi di gruppi del pool di utenti che includono l'utente come membro.

iss

Il provider di identità che ha emesso il token. L'attestazione ha il seguente formato.

`https://cognito-idp.<Region>.amazonaws.com/<your user pool ID>`

client_id

Il client dell'app del pool di utenti che ha autenticato l'utente. Amazon Cognito restituisce lo stesso valore nell'attestazione `aud` del token ID.

origin_jti

Un identificatore di revoca del token associato al token di aggiornamento dell'utente. Amazon Cognito fa riferimento all'`origin_jti`affermazione quando verifica se hai revocato il token dell'utente con l'operazione Endpoint Revoke o l'RevokeTokenAPI. Quando revochi un token, Amazon Cognito invalida tutti i token di accesso e ID con lo stesso valore `origin_jti`.

token_use

Lo scopo previsto del token. In un token di accesso, il valore è `access`.

scope

Un elenco di ambiti OAuth 2.0 che definiscono l'accesso fornito dal token. Un token dell'Endpoint Token può contenere tutti gli ambiti supportati dal client dell'app. Un token dell'accesso API di Amazon Cognito contiene solo l'ambito `aws.cognito.signin.user.admin`.

auth_time

L'ora di fine dell'autenticazione dell'utente, in formato Unix.

exp

L'ora di scadenza del token dell'utente, in formato Unix.

`iat`

L'ora di emissione del token dell'utente da parte di Amazon Cognito, in formato Unix.

`jti`

L'identificatore univoco del JWT.

`username`

Il nome utente nel pool di utenti.

Firma del token di accesso

La firma del token di accesso viene calcolata in base all'intestazione e al payload del token JWT. Se utilizzi il token al di fuori di un'applicazione nelle API Web, devi verificare sempre questa firma prima di accettare il token. Per ulteriori informazioni, consulta Verifica di un JSON Web Token.

Utilizzo del token di aggiornamento

Puoi utilizzare il token di aggiornamento per recuperare nuovi ID e token di accesso. Di default, il token di aggiornamento scade 30 giorni dopo l'accesso dell'utente dell'app al bacino d'utenza. Quando crei un'applicazione per il bacino d'utenza, puoi impostare la scadenza del token di aggiornamento dell'applicazione su qualsiasi valore compreso tra 60 minuti e 10 anni.

Mobile SDK for iOS, Mobile SDK for Android, Amplify for iOS, Android e Flutter aggiornano automaticamente gli ID e i token di accesso se è presente un token di aggiornamento valido (non scaduto). Gli ID e i token di accesso hanno una validità residua minima di 2 minuti. Se il token di aggiornamento è scaduto, l'utente dell'applicazione dovrà eseguire nuovamente l'autenticazione per accedere di nuovo al bacino d'utenza. Se il valore minimo per il token di accesso e il token ID è impostato su 5 minuti e si utilizza l'SDK, il token di aggiornamento verrà utilizzato costantemente per ripristinare i nuovi token ID e token di accesso. Sarà possibile visualizzare il comportamento previsto impostando un valore minimo di 7 minuti anziché di 5.

L'account dell'utente di per sé non scade, a condizione che l'utente abbia effettuato l'accesso almeno una volta prima del limite di tempo di `UnusedAccountValidityDays` per nuovi account.

Ottenere nuovi token di accesso e identità con un token di aggiornamento

Utilizza l'API o l'interfaccia utente ospitata per avviare l'autenticazione dei token di aggiornamento.

Per utilizzare il token di aggiornamento per ottenere un nuovo ID e accedere ai token con l'API dei pool di utenti, utilizza le AdminInitiateAuthoperazioni o API. InitiateAuth Invia `REFRESH_TOKEN_AUTH` per il parametro `AuthFlow`. Nella proprietà `AuthParameters` di `AuthFlow`, passa il token di aggiornamento dell'utente come valore di `"REFRESH_TOKEN"`. Amazon Cognito restituisce ID e token di accesso nuovi dopo che la richiesta API ha superato tutte le sfide.

> **Note**
>
> Per utilizzare l'API dei pool di utenti di Amazon Cognito per aggiornare i token per un utente dell'interfaccia utente ospitata, genera una richiesta `InitiateAuth`.

Puoi anche inviare token di aggiornamento a Endpoint Token in un pool di utenti in cui hai configurato un dominio. Nel corpo della richiesta, includi un valore `grant_type` di `refresh_token` e un valore `refresh_token` del token di aggiornamento dell'utente.

Revoca dei token di aggiornamento

È possibile revocare i token di aggiornamento che appartengono a un utente. Per ulteriori informazioni sulla revoca dei token, consulta Revoca dei token.

> **Note**
>
> La revoca del token di aggiornamento comporta la revoca di tutti i token ID e di accesso emessi da Amazon Cognito a seguito delle richieste di aggiornamento con tale token.

Quando revochi tutti i token dell'utente tramite le operazioni API `GlobalSignOut` e `AdminUserGlobalSignOut`, gli utenti possono disconnettersi da tutti i dispositivi a cui sono attualmente collegati. Dopo che l'utente viene disconnesso, si verifica quanto segue.

- Il token di aggiornamento dell'utente non può ottenere nuovi token per l'utente.
- Il token di accesso dell'utente non può effettuare richieste API autorizzate dal token.
- L'utente deve effettuare nuovamente l'autenticazione per ottenere nuovi token. Poiché i cookie della sessione dell'interfaccia utente ospitata non scadono automaticamente, l'utente può eseguire nuovamente l'autenticazione con un cookie di sessione, senza richiedere credenziali aggiuntive. Dopo che gli utenti dell'interfaccia utente ospitata sono stati disconnessi, reindirizzali al Endpoint Logout, in cui Amazon Cognito cancellerà il relativo cookie di sessione.

Con i token di aggiornamento, puoi mantenere le sessioni degli utenti nella tua app a lungo. Nel tempo, i tuoi utenti potrebbero richiedere di rimuovere l'autorizzazione da alcuni dispositivi a cui hanno effettuato l'accesso, aggiornando continuamente la loro sessione. Per disconnettere l'utente da un singolo dispositivo, revoca il relativo token di aggiornamento. Quando l'utente desidera disconnettersi da tutte le sessioni autenticate, genera una richiesta API. GlobalSignOut La tua app può presentare all'utente una scelta come Esci da tutti i dispositivi. `GlobalSignOut` accetta un token di accesso valido, inalterato, non scaduto e non revocato, di un utente. Poiché questa API è autorizzata dal token, un utente non può utilizzarla per avviare la disconnessione di un altro utente.

Tuttavia, puoi generare una richiesta AdminUserGlobalSignOutAPI che autorizzi con le tue AWS credenziali per disconnettere qualsiasi utente da tutti i suoi dispositivi. L'applicazione di amministrazione deve richiamare questa operazione API con le credenziali AWS dello sviluppatore e passare l'ID del pool di utenti e il nome utente dell'utente come parametri. L'API `AdminUserGlobalSignOut` è in grado di disconnettere qualsiasi utente nel bacino d'utenza.

Per ulteriori informazioni sulle richieste che è possibile autorizzare con AWS credenziali o con il token di accesso di un utente, vedere. Operazioni API autenticate e non autenticate per pool di utenti di Amazon Cognito

Revoca dei token

Puoi revocare un token di aggiornamento per un utente utilizzando l'API. AWS Quando si revoca un token di aggiornamento, tutti i token di accesso precedentemente emessi da tale token di aggiornamento diventano non validi. Gli altri token di aggiornamento rilasciati all'utente non sono interessati da questa operazione.

> **Note**
>
> I token JWT includono una firma e una scadenza indipendenti assegnate al momento della creazione. I token revocati non possono essere utilizzati con chiamate API Amazon Cognito che richiedono un token. Tuttavia, i token revocati saranno comunque validi se vengono verificati utilizzando una qualsiasi libreria JWT che verifica la firma e la scadenza del token.

È possibile revocare un token di aggiornamento per il client di un bacino d'utenza se la revoca di token è abilitata. Quando crei un nuovo client del bacino d'utenza, la revoca dei token è attivata di default.

Abilitazione della revoca dei token

Prima di poter revocare un token per un client del bacino d'utenza esistente, è necessario abilitare la revoca del token. È possibile abilitare la revoca del token per i client del pool di utenti esistenti utilizzando o l' AWS CLI API. AWS Per fare ciò, chiama il comando `aws cognito-idp describe-user-pool-client` della CLI o l'operazione API `DescribeUserPoolClient` per recuperare le impostazioni correnti dal client app. Quindi, chiama il comando `aws cognito-idp update-user-pool-client` della CLI o l'operazione API `UpdateUserPoolClient`. Includi le impostazioni correnti dal client app e imposta il parametro `EnableTokenRevocation` su `true`.

Quando si crea un nuovo client con pool di utenti utilizzando l' AWS Management Console, la o l' AWS API AWS CLI, la revoca dei token è abilitata per impostazione predefinita.

Dopo aver abilitato la revoca dei token, nuove richieste vengono aggiunte nei token web JSON di Amazon Cognito. Le attestazioni `origin_jti` e `jti` vengono aggiunte ai token di accesso e ID. Queste attestazioni aumentano le dimensioni dei token ID e di accesso del client dell'app.

Per creare o modificare un client di app con la revoca dei token abilitata, includi il seguente parametro nella tua richiesta CreateUserPoolClientto UpdateUserPoolClientAPI.

```
"EnableTokenRevocation": true
```

Revoca di un token

È possibile revocare un token di aggiornamento utilizzando una richiesta RevokeTokenAPI, ad esempio con il comando `aws cognito-idp revoke-token` CLI. Puoi anche revocare i token usando il Endpoint Revoke. Questo endpoint diventa disponibile dopo l'aggiunta di un dominio al bacino d'utenza. Puoi utilizzare l'endpoint di revoca su un dominio ospitato Amazon Cognito o sul tuo dominio personalizzato.

> **Note**
>
> La richiesta per revocare un token di aggiornamento deve includere l'ID client utilizzato per ottenerlo.

Di seguito è riportato il corpo di una richiesta API `RevokeToken` di esempio.

```
{
```

```
    "ClientId": "1example23456789",
    "ClientSecret": "abcdef123456789ghijklexample",
    "Token": "eyJjdHkiOiJKV1QiEXAMPLE"
}
```

Di seguito è riportato un esempio di richiesta cURL all'endpoint `/oauth2/revoke` di un pool di utenti con un dominio personalizzato.

```
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'operazione `RevokeToken` e l'endpoint `/oauth2/revoke` non richiedono alcuna autorizzazione aggiuntiva a meno che il client dell'app non disponga di un segreto del client.

Verifica di un JSON Web Token

Queste fasi descrivono la verifica del token JWT (JSON Web Token) di un bacino d'utenza.

Argomenti

- Prerequisiti
- Convalida i token con aws-jwt-verify
- Comprensione e ispezione dei token

Prerequisiti

La tua libreria, SDK o il framework del software potrebbero già gestire le attività in questa sezione. AWS Gli SDK forniscono strumenti per la gestione e la gestione dei token del pool di utenti di Amazon Cognito nella tua app. AWS Amplify include funzioni per recuperare e aggiornare i token Amazon Cognito.

Per ulteriori informazioni, consulta le pagine seguenti.

- Integrazione dell'autenticazione e dell'autorizzazione di Amazon Cognito con app web e mobili
- Esempi di codice per Amazon Cognito Identity Provider che utilizza SDK AWS
- Advanced workflows (Flussi di lavoro avanzati) in Amplify Dev Center (Centro per sviluppatori Amplify)

Per la decodifica e la verifica di un token Web JSON (JWT) sono disponibili numerose librerie. Queste librerie sono utili se vuoi gestire i token manualmente per l'elaborazione dell'API lato server oppure se stai usando altri linguaggi di programmazione. Consulta l'elenco di librerie OpenID Foundation per l'operatività con i token JWT.

Convalida i token con aws-jwt-verify

In un'app Node.js, AWS consiglia alla aws-jwt-verifylibreria di convalidare i parametri nel token che l'utente passa all'app. Con `aws-jwt-verify` puoi popolare `CognitoJwtVerifier` con i valori delle attestazioni che desideri verificare per uno o più pool di utenti. Di seguito sono riportati alcuni valori che possono essere verificati.

- Verifica che i token di accesso o ID non siano malformati o scaduti e abbiano una firma valida.
- Verifica che i token di accesso provengano dai pool di utenti e dai client dell'app corretti.
- Verifica che le attestazioni del token di accesso contengano gli ambiti OAuth 2.0 corretti.
- Verifica che le chiavi che hanno firmato i token ID e di accesso corrispondano a una chiave di firma `kid` dell'URI JWKS dei pool di utenti.

  L'URI JWKS contiene le informazioni pubbliche sulla chiave privata usata per la firma del token dell'utente. Puoi trovare l'URI JWKS del tuo pool di utenti all'indirizzo `https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json`.

Per ulteriori informazioni e codice di esempio che è possibile utilizzare in un'app Node.js o in un programma di AWS Lambda autorizzazione, vedere aws-jwt-verifysu. GitHub

Comprensione e ispezione dei token

Prima di integrare l'ispezione dei token nell'app occorre considerare come Amazon Cognito assembla i JWT. Recupera i token di esempio dal pool di utenti, decodificali ed esaminali accuratamente per comprenderne le caratteristiche e determinare cosa vuoi verificare e quando. Ad esempio, potresti voler esaminare l'appartenenza al gruppo in uno scenario e gli ambiti in un altro.

Le sezioni seguenti descrivono un processo per ispezionare manualmente i JWT di Amazon Cognito durante la preparazione dell'app.

Conferma della struttura del JWT

Un token Web JSON (JWT) include tre sezioni separate con il delimitatore `.` (punto).

Header

L'ID della chiave (`kid`) e l'algoritmo RSA (`alg`) utilizzati da Amazon Cognito per firmare il token. Amazon Cognito firma i token con un `alg` di `RS256`.

Payload

Le attestazioni del token. In un token ID, le attestazioni includono gli attributi dell'utente e le informazioni sul pool di utenti (`iss`) e sul client dell'app (`aud`). In un token di accesso, il payload include gli ambiti, l'appartenenza al gruppo, il pool di utenti come `iss` e il client dell'app come `client_id`.

Firma

La firma non è decodificabile in base64 come l'intestazione e il payload. È un identificatore RSA256 derivato da una chiave di firma e dai parametri che puoi osservare all'URI JWKS.

L'intestazione e il payload sono JSON con codifica base64. Puoi identificarli tramite i caratteri di apertura `eyJ` che si decodificano nel carattere iniziale `{`. Se l'utente presenta all'app un JWT con codifica base64 che non è nel formato `[JSON Header].[JSON Payload].[Signature]`, non è un token di Amazon Cognito valido e puoi eliminarlo.

Convalida del JWT

La firma JWT è una combinazione con hash di intestazione e payload. Amazon Cognito genera due coppie di chiavi di crittografia RSA per ogni bacino d'utenza. Una chiave privata firma i token di accesso e l'altra i token ID.

Per verificare la firma di un token JWT

1. Decodifica il token ID.

   Anche OpenID Foundation mantiene un elenco di librerie per l'utilizzo di token JWT.

   Puoi anche usarlo AWS Lambda per decodificare il pool di utenti JWT. Per ulteriori informazioni, consulta Decodificare e verificare utilizzando i token Amazon Cognito JWT. AWS Lambda
2. Confronta l'ID della chiave (`kid`) locale con il `kid` pubblico.

   a. Scarica e archivia la chiave JWK (JSON Web Key) pubblica corrispondente per il tuo bacino d'utenza. È disponibile come parte di un set JWKS (JSON Web Key Set). Puoi individuarlo creando il seguente URI `jwks_uri` per l'ambiente:

```
https://cognito-idp.<Region>.amazonaws.com/<userPoolId>/.well-known/jwks.json
```

Per ulteriori informazioni su JWK e JWKS, consulta l'articolo su JSON Web Key (JWK).

> **Note**
>
> Amazon Cognito può alternare le chiavi di firma nel pool di utenti. Come procedura consigliata, memorizza nella cache le chiavi pubbliche dell'app utilizzando il `kid` come chiave di cache e aggiorna periodicamente la cache. Confronta il `kid` nei token che la tua app riceve nella cache.
> Se ricevi un token con l'emittente corretto ma un diverso `kid`, è possibile che Amazon Cognito abbia alternato la chiave di firma. Aggiorna la cache dall'endpoint `jwks_uri` del pool di utenti.

Questo è un file `jwks.json` di esempio:

```
{
 "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 chiave (**`kid`**)

Il `kid` è un suggerimento che indica quale chiave è stata utilizzata per proteggere la JSON Web Signature (JWS) del token.

Algoritmo (**`alg`**)

Il parametro di intestazione `alg` rappresenta l'algoritmo di crittografia utilizzato per proteggere il token ID. I bacini d'utenza utilizzano un algoritmo crittografico RS256, ovvero una firma RSA con SHA-256. Per ulteriori informazioni su RSA, consulta crittografia RSA.

Tipo di chiavi (**`kty`**)

Il parametro `kty` identifica la famiglia di algoritmi di crittografia usata con la chiave, come "RSA" in questo esempio.

Esponente RSA (**`e`**)

Il parametro `e` contiene il valore dell'esponente per la chiave pubblica RSA. È rappresentato come valore con codifica Base64urlUInt.

Modulo RSA (**`n`**)

Il parametro `n` contiene il valore del modulo per la chiave pubblica RSA. È rappresentato come valore con codifica Base64urlUInt.

Utilizza (**`use`**)

Il parametro `use` descrive l'uso previsto della chiave pubblica. Per questo esempio, il valore `use` value `sig` rappresenta la firma.

b. Cerca la JSON Web Key pubblica per un `kid` che corrisponde al `kid` del JWT.

3. Utilizza una libreria JWT per confrontare la firma dell'emittente con la firma nel token. La firma dell'emittente è derivata dalla chiave pubblica (modulo RSA "`n`") del `kid` in jwks.json che corrisponde al token `kid`. Per prima cosa, potrebbe essere necessario convertire il formato JWK in PEM. Questo esempio specifica il JWT e la JWK e utilizza la libreria Node.js jsonwebtoken per verificare la firma del 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) {
});
```

Verifica delle attestazioni

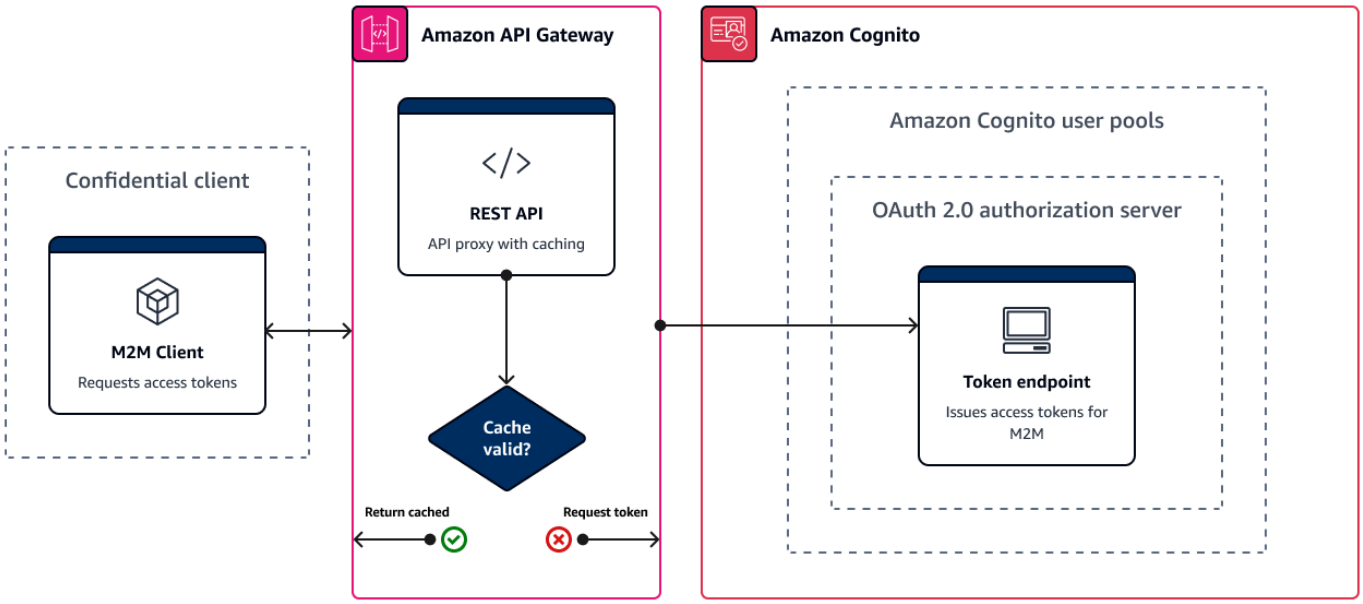
Per verificare le attestazioni JWT

1. Con uno dei seguenti metodi, verifica che il token non sia scaduto.
   a. Decodifica il token e confronta la richiesta `exp` con l'ora corrente.
   b. Se il tuo token di accesso include un `aws.cognito.signin.user.admin` reclamo, invia una richiesta a un'API come. GetUser Le richieste API di autorizzazione con un token di accesso restituiscono un errore se il token è scaduto.
   c. Presenta il tuo token di accesso in una richiesta a Endpoint UserInfo. La tua richiesta restituisce un errore se il token è scaduto.
2. L'attestazione `aud` in un token ID e l'attestazione `client_id` in un token di accesso deve corrispondere all'ID client di app creato nel bacino d'utenza di Amazon Cognito.
3. L'attestazione dell'approvatore (`iss`) deve corrispondere al bacino d'utenza. Ad esempio, un bacino d'utenza creato nella regione `us-east-1` avrà un valore `iss` di:

   `https://cognito-idp.us-east-1.amazonaws.com/`*`<userpoolID>`*.
4. Controlla l'attestazione `token_use`.
   - Se accetti il token di accesso solo nelle operazioni dell'API Web, il valore deve essere `access`.
   - Se utilizzi solo il token ID, il valore deve essere `id`.
   - Se si usano sia ID sia token di accesso, l'attestazione `token_use` deve essere `id` o `access`.

Ora è possibile considerare attendibili le attestazioni all'interno del token.

Caching dei token



L'app deve completare una delle seguenti richieste ogni volta che desideri ottenere un nuovo token web JSON (JWT).

- Richiesta di una concessione di credenziali client o codici di autorizzazione dall'Endpoint Token.
- Richiesta di una concessione implicita dall'interfaccia utente ospitata.
- Autentica un utente locale in una richiesta API Amazon Cognito come. InitiateAuth

Puoi configurare il pool di utenti in modo da impostare la scadenza dei token entro un valore definito in minuti, ore o giorni. Per garantire le prestazioni e la disponibilità della tua app, usa i token Amazon Cognito fino alla scadenza, trascorsa la quale puoi recuperarne di nuovi. Una soluzione di caching creata per l'app mantiene disponibili i token e impedisce ad Amazon Cognito di rifiutare le richieste quando la frequenza delle richieste è troppo alta. Un'app lato client deve archiviare i token in una cache di memoria. Un'app lato server può aggiungere un meccanismo di caching crittografato per archiviare i token.

Quando il tuo pool di utenti genera un volume elevato di utenti o machine-to-machine attività, potresti incontrare i limiti imposti da Amazon Cognito sul numero di richieste di token che puoi effettuare. Per ridurre il numero di richieste inviate agli endpoint Amazon Cognito, è possibile archiviare e riutilizzare in modo sicuro i dati di autenticazione o implementare backoff esponenziale e nuovi tentativi.

I dati di autenticazione provengono da due classi di endpoint. Gli endpoint OAuth 2.0 di Amazon Cognito includono l'endpoint token, che gestisce le credenziali client e le richieste di codice di autorizzazione dell'interfaccia utente ospitata. Gli endpoint del servizio rispondono a richieste API come `InitiateAuth` e `RespondToAuthChallenge`. Ogni tipo di richiesta è caratterizzato da un limite specifico. Per ulteriori informazioni sui limiti, consulta Quote in Amazon Cognito.

Memorizzazione nella cache dei token di machine-to-machine accesso con Amazon API Gateway

Con il caching dei token di Gateway API, l'app può eseguire il ridimensionamento in risposta a eventi che superano il limite predefinito per la frequenza delle richieste degli endpoint OAuth di Amazon Cognito.

Puoi memorizzare nella cache i token di accesso in modo che l'app richieda un nuovo token di accesso solo se un token memorizzato nella cache è scaduto. In caso contrario, l'endpoint di caching restituisce un token dalla cache. Ciò evita una chiamata aggiuntiva a un endpoint dell'API Amazon Cognito. Quando si utilizza Gateway Amazon API come proxy per l'Endpoint Token, l'API risponde alla maggior parte delle richieste che altrimenti contribuirebbero alla quota di richieste, evitando richieste non riuscite risultanti dalla limitazione della frequenza.

La seguente soluzione basata su Gateway API offre un'implementazione a bassa latenza, a bassa presenza di codice/senza codice del caching dei token. Le API di Gateway API sono crittografate in transito e, facoltativamente, quando sono inattive. Una cache API Gateway è ideale per la concessione delle credenziali del client OAuth 2.0, un tipo di concessione spesso ad alto volume che produce token di accesso per l'autorizzazione e sessioni di microservizi. machine-to-machine In un evento come un aumento del traffico che causa la scalabilità orizzontale dei microservizi, è possibile che molti sistemi utilizzino le stesse credenziali client a un volume che supera il limite di frequenza delle richieste del pool di utenti o del client dell'app. AWS Per preservare la disponibilità delle app e la bassa latenza, una soluzione di caching è la best practice consigliata in tali scenari.

In questa soluzione si definisce una cache nell'API per archiviare un token di accesso distinto per ogni combinazione di ambiti OAuth e client dell'app che si desidera richiedere nell'app. Quando l'app effettua una richiesta corrispondente alla chiave della cache, l'API risponde con un token di accesso emesso da Amazon Cognito alla prima richiesta corrispondente alla chiave della cache. Alla scadenza della chiave, l'API inoltra la richiesta all'endpoint del token e memorizza nella cache un nuovo token di accesso.

> **Note**
>
> La durata della chiave della cache deve essere inferiore alla durata del token di accesso del client dell'app.

La chiave della cache è una combinazione degli ambiti OAuth richiesti nel parametro URL `scope` e nell'intestazione `Authorization` nella richiesta. L'intestazione `Authorization` contiene l'ID del client dell'app e il segreto client. Non è necessario implementare una logica aggiuntiva nell'app per implementare questa soluzione. È solo necessario aggiornare la configurazione per modificare il percorso dell'endpoint del token del pool di utenti.

Puoi anche implementare la memorizzazione nella cache dei token con for Redis. ElastiCache Per un controllo granulare con le policy AWS Identity and Access Management (IAM), valuta l'ipotesi di utilizzare una cache Amazon DynamoDB.

> **Note**
>
> Il caching in Gateway API è soggetto a costi aggiuntivi. Per maggiori dettagli, consulta la pagina relativa ai prezzi.

Per configurare un proxy di caching con Gateway API

1. Apri la Console Gateway API e crea una REST API.
2. In Resources (Risorse), crea un metodo POST.
    a. In Integration type (Tipo di integrazione), scegli HTTP.
    b. Seleziona Use HTTP proxy integration (Utilizza integrazione proxy HTTP).
    c. In Endpoint URL (URL endpoint), immetti un URL nel formato `https://<your user pool domain>/oauth2/token`.
3. In Resources (Risorse), configura la chiave della cache.
    a. In Method request (Richiesta metodo), modifica la richiesta del metodo POST.
    b. Imposta il parametro `scope` e l'intestazione `Authorization` come chiave di caching.
        i. Aggiungi una stringa di query in URL query string parameters (Parametri della stringa di query URL) e scegli Caching per la stringa `scope`.

ii. Aggiungi un'intestazione in HTTP request headers (Intestazioni di richiesta HTTP) e scegli Caching per l'intestazione `Authorization`.

4. In Stages (Fasi), configura il caching.
   a. Scegliere la fase da modificare.
   b. In Settings (Impostazioni), seleziona Enable API cache (Abilita API cache).
   c. Scegli un valore in Cache capacity (Capacità cache).
   d. Scegli una cache time-to-live (TTL) di almeno 3600 secondi.
   e. Deseleziona la casella di controllo Richiedi autorizzazione.
5. In Stages (Fasi), annota il valore visualizzato in Invoke URL (Richiama URL).
6. Aggiorna la tua app per le richieste di token POST in base al valore di Invoke URL (Richiama URL) della tua API anziché in base all'endpoint `/oauth2/token` del pool di utenti.

Accesso alle risorse dopo una corretta autenticazione del bacino d'utenza

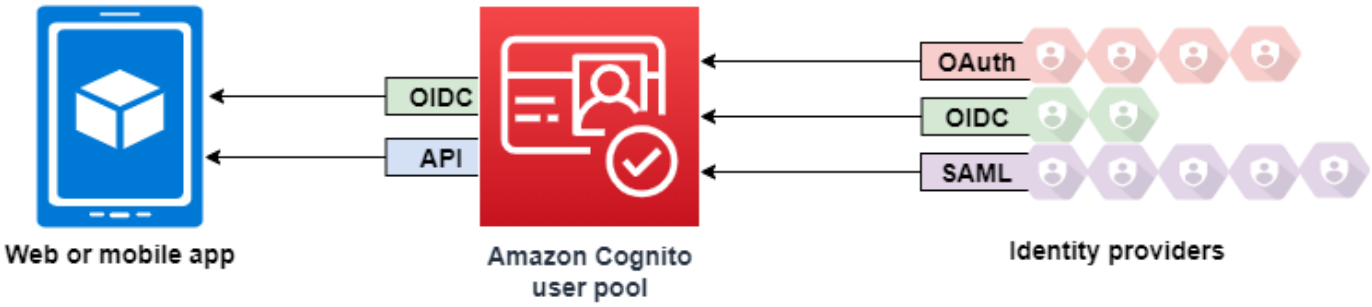
Gli utenti della tua app possono accedere direttamente tramite un pool di utenti oppure possono federarsi tramite un provider di identità (IdP) di terze parti. Il pool di utenti gestisce il sovraccarico di gestione dei token restituiti dall'accesso social tramite Facebook, Google, Amazon e Apple e da OpenID Connect (OIDC) e SAML. IdPs Per ulteriori informazioni, consulta Utilizzo di token con bacini d'utenza.

Dopo una corretta autenticazione, l'app riceverà i token del bacino d'utenza da Amazon Cognito. Puoi utilizzare i token del pool di utenti per:

- Recupera AWS le credenziali che autorizzano le richieste di risorse applicative in Amazon Servizi AWS DynamoDB e Amazon S3.
- Fornisci una prova di autenticazione temporanea e revocabile.
- Inserisci i dati di identità in un profilo utente nella tua app.
- Autorizza le modifiche al profilo dell'utente che ha effettuato l'accesso nella directory del pool di utenti.
- Autorizza le richieste di informazioni sugli utenti con un token di accesso.
- Autorizza le richieste ai dati che si trovano dietro API esterne protette dall'accesso con token di accesso.

- Autorizza l'accesso agli asset applicativi archiviati sul client o sul server con Amazon Verified Permissions.

Per ulteriori informazioni, consulta Flusso di autenticazione del bacino d'utenza e Utilizzo di token con bacini d'utenza.



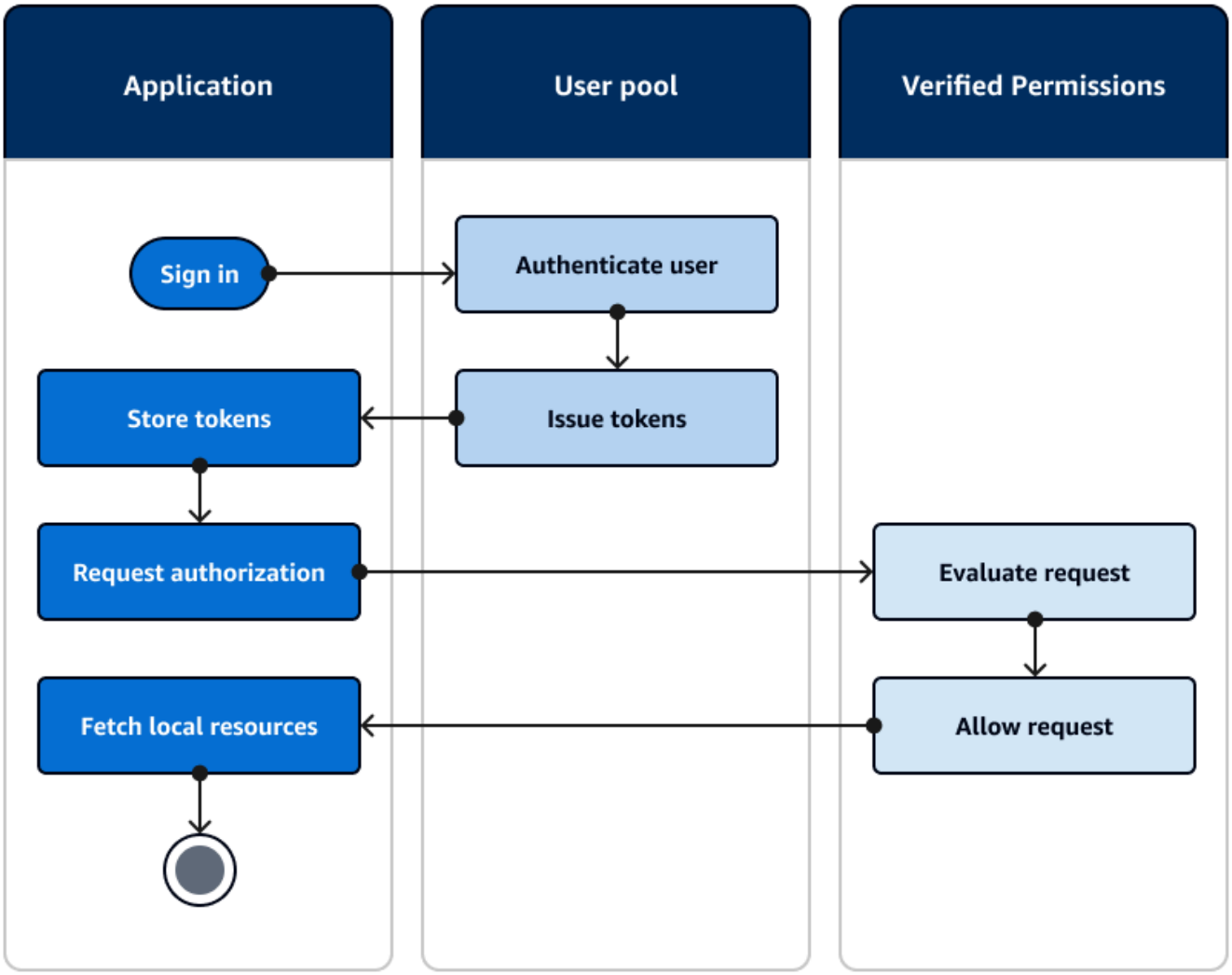
Argomenti

- Autorizzazione dell'accesso alle risorse del client o del server con Amazon Verified Permissions
- Accesso alle risorse con API Gateway dopo l'accesso
- Accesso Servizi AWS tramite un pool di identità dopo l'accesso

Autorizzazione dell'accesso alle risorse del client o del server con Amazon Verified Permissions

La tua app può trasferire i token da un utente che ha effettuato l'accesso ad Amazon Verified Permissions. Verified Permissions è un servizio scalabile e dettagliato di gestione e autorizzazione delle autorizzazioni per le applicazioni personalizzate che hai creato. Un pool di utenti di Amazon Cognito può fungere da fonte di identità per un archivio di policy di Autorizzazioni verificate. Verified Permissions prende decisioni di autorizzazione per le azioni e le risorse richieste, ad `GetPhoto` esempio`premium_badge.png`, a partire dal principale e dai relativi attributi nei token del pool di utenti.

Il diagramma seguente mostra come l'applicazione può passare il token di un utente a Verified Permissions in una richiesta di autorizzazione.



Inizia a usare Amazon Verified Permissions

Dopo aver integrato il tuo pool di utenti con Autorizzazioni verificate, ottieni una fonte centrale di autorizzazione granulare per tutte le tue app Amazon Cognito. Ciò elimina la necessità di una logica di sicurezza granulare che altrimenti dovresti codificare e replicare tra tutte le tue app. Per ulteriori informazioni sull'autorizzazione con autorizzazioni verificate, consulta. Autorizzazione con Amazon Verified Permissions

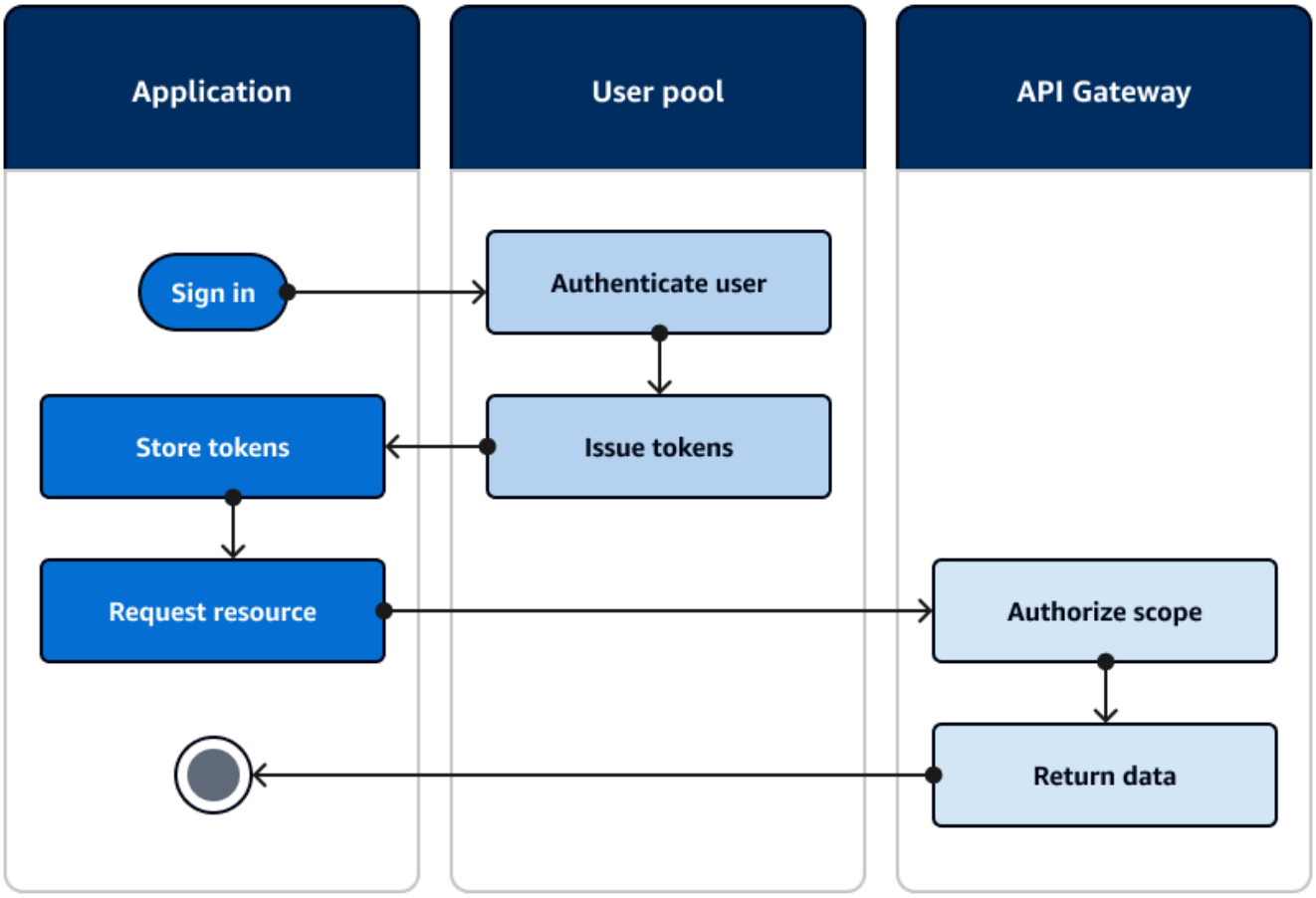
Le richieste di autorizzazione per le autorizzazioni verificate richiedono credenziali AWS . È possibile implementare alcune delle seguenti tecniche per applicare in modo sicuro le credenziali alle richieste di autorizzazione.

- Gestisci un'applicazione web in grado di archiviare segreti nel backend del server.
- Acquisisci credenziali autenticate del pool di identità.
- Effettua il proxy delle richieste degli utenti tramite un' access-token-authorized API e aggiungi AWS le credenziali alla richiesta.

Accesso alle risorse con API Gateway dopo l'accesso

Un uso comune dei token dei pool di utenti di Amazon Cognito consiste nell'autorizzare le richieste a un'API REST di API Gateway. Gli ambiti OAuth 2.0 nei token di accesso possono autorizzare un metodo e un percorso, come for. `HTTP GET /app_assets` I token ID possono fungere da autenticazione generica per un'API e passare gli attributi utente al servizio di backend. API Gateway offre opzioni di autorizzazione personalizzate aggiuntive come gli autorizzatori JWT per le API HTTP e gli autorizzatori Lambda che possono applicare una logica più dettagliata.

Il diagramma seguente illustra un'applicazione che sta ottenendo l'accesso a un'API REST con gli ambiti OAuth 2.0 in un token di accesso.



L'app deve raccogliere i token dalle sessioni autenticate e aggiungerli come token portatori a un'intestazione della richiesta. `Authorization` Configura l'autorizzatore che hai configurato per l'API, il percorso e il metodo per valutare il contenuto dei token. API Gateway restituisce i dati solo se la richiesta soddisfa le condizioni impostate per l'autorizzatore.

Alcuni modi potenziali in cui l'API API Gateway può approvare l'accesso da un'applicazione sono:

- Il token di accesso contiene l'ambito OAuth 2.0 corretto. L'autorizzazione dei pool di utenti di Amazon Cognito per un'API REST è un'implementazione comune con una bassa barriera all'ingresso. Puoi anche valutare il corpo, i parametri della stringa di query e le intestazioni di una richiesta a questo tipo di autorizzazione.
- Il token ID è valido e non è scaduto. Quando passi un token ID a un autorizzatore Amazon Cognito, puoi eseguire un'ulteriore convalida del contenuto del token ID sul tuo server delle applicazioni.
- Un gruppo, una dichiarazione, un attributo o un ruolo in un token di accesso o ID soddisfa i requisiti definiti in una funzione Lambda. Un autorizzatore Lambda analizza il token nell'intestazione della

richiesta e lo valuta per una decisione di autorizzazione. Puoi creare una logica personalizzata nella tua funzione o effettuare una richiesta API ad Amazon Verified Permissions.

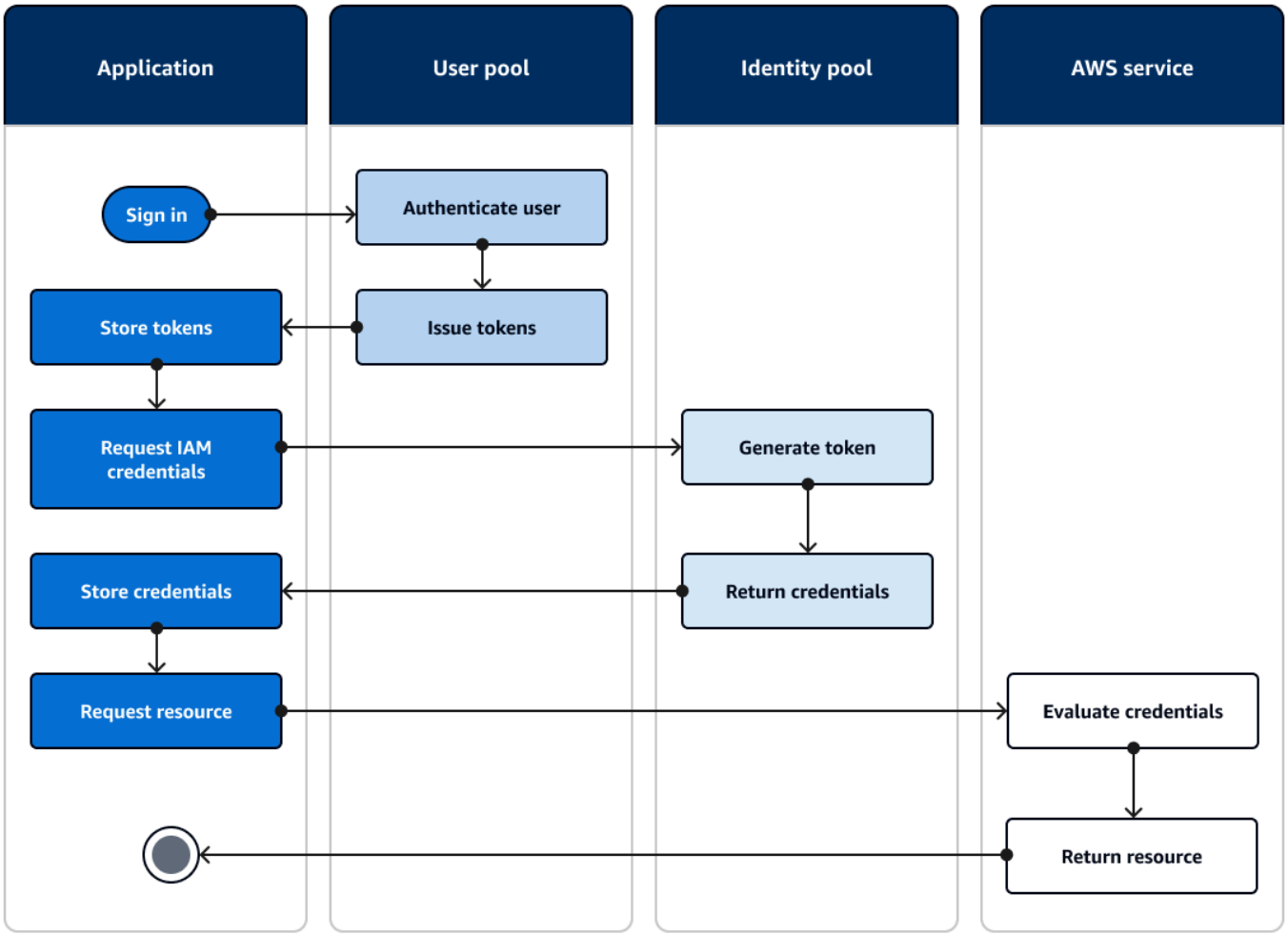
Puoi anche autorizzare le richieste a un'API AWS AppSync GraphQL con token provenienti da un pool di utenti.

Accesso Servizi AWS tramite un pool di identità dopo l'accesso

Dopo aver effettuato l'accesso con un pool di utenti, gli utenti possono accedere Servizi AWS con credenziali API temporanee emesse da un pool di identità.

La tua app web o mobile riceve token da un pool di utenti. Quando configuri il tuo pool di utenti come provider di identità per il tuo pool di identità, il pool di identità scambia i token con credenziali temporanee AWS . Queste credenziali possono essere applicate ai ruoli IAM e alle relative politiche che consentono agli utenti di accedere a un set limitato di risorse. AWS Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità (identità federate).

Il diagramma seguente mostra come un'applicazione accede a un pool di utenti, recupera le credenziali del pool di identità e richiede una risorsa da un. Servizio AWS



Puoi utilizzare le credenziali del pool di identità per:

- Effettua richieste di autorizzazione dettagliate ad Amazon Verified Permissions con le credenziali del tuo utente.
- Connettiti a un'API REST di Amazon API Gateway o a un'API AWS AppSync GraphQL che autorizza le connessioni con IAM.
- Connettiti a un backend di database come Amazon DynamoDB o Amazon RDS che autorizza le connessioni con IAM.
- Recupera gli asset applicativi da un bucket Amazon S3.
- Avvia una sessione con un desktop WorkSpaces virtuale Amazon.

I pool di identità non funzionano esclusivamente all'interno di una sessione autenticata con un pool di utenti. Accettano inoltre l'autenticazione direttamente da provider di identità di terze parti e possono generare credenziali per utenti ospiti non autenticati.

Per ulteriori informazioni sull'utilizzo dei pool di identità insieme ai gruppi di pool di utenti per controllare l'accesso alle AWS risorse, consulta e. Aggiunta di gruppi a un bacino d'utenza Utilizzo del controllo degli accessi basato su ruoli Inoltre, per ulteriori informazioni sui pool di identità e AWS Identity and Access Management, vedereConcetti del pool di identità.

Configurazione di un pool di utenti con AWS Management Console

Crea un bacino d'utenza di Amazon Cognito e prendi nota dell'ID bacino d'utenza e dell'ID client app per ciascuna delle app client. Per ulteriori informazioni sulla creazione di bacini d'utenza, consulta l'articolo Nozioni di base sui bacini d'utenza.

Configurazione di un pool di identità con AWS Management Console

La procedura seguente descrive come utilizzare per AWS Management Console integrare un pool di identità con uno o più pool di utenti e app client.

Per aggiungere un gestore dell'identità digitale al pool di utenti Amazon Cognito

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Pool di utenti Amazon Cognito.
5. Inserisci un ID del pool di utenti e un ID client dell'app.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    a. Puoi assegnare agli utenti di quell'IdP il ruolo predefinito che hai impostato quando hai configurato il ruolo Autenticato oppure puoi scegliere il ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con attestazione preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

        i. Se hai scelto Scegli il ruolo con le regole, inserisci la dichiarazione di origine dall'autenticazione dell'utente, l'operatore che desideri utilizzare per confrontare l'affermazione con la regola, il valore che determinerà la corrispondenza con questa

scelta di ruolo e il ruolo che desideri assegnare quando l'assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Se hai scelto Choose role with preferred_role claim in token, Amazon Cognito emette le credenziali per il ruolo nel claim dell'utente. `cognito:preferred_role` Se non è presente alcuna dichiarazione di ruolo preferito, Amazon Cognito emette le credenziali in base alla risoluzione del ruolo.

b. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   - Per non applicare alcun tag principale, scegli Inattivo.
   - Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
   - Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Integrazione di un bacino d'utenza con un pool di identità

Dopo l'autenticazione dell'utente dell'app, aggiungi il relativo token di identità alla mappa degli accessi nel fornitore di credenziali. Il nome del provider dipenderà dal tuo ID bacino d'utenza di Amazon Cognito. avrà la struttura seguente:

```
cognito-idp.<region>.amazonaws.com/<YOUR_USER_POOL_ID>
```

<region>Puoi ricavare il valore per dall'ID del pool di utenti. Ad esempio, se l'ID del pool di utenti è`us-east-1_EXAMPLE1`, allora <region>è`us-east-1`. Se l'ID del pool di utenti è`us-west-2_EXAMPLE2`, allora <region>è`us-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(serviceConfigura
 userPoolConfiguration: userPoolConfiguration, forKey: "UserPool")
let pool = AWSCognitoIdentityUserPool(forKey: "UserPool")
let credentialsProvider = AWSCognitoCredentialsProvider(regionType: .USEast1,
 identityPoolId: "YOUR_IDENTITY_POOL_ID", identityProviderManager:pool)
```

Utilizzo delle caratteristiche di sicurezza dei pool di utenti di Amazon Cognito

Puoi aggiungere l'autenticazione a più fattori o MFA (Multi-Factor Authentication) a un bacino d'utenza per proteggere l'identità dei tuoi utenti. L'autenticazione MFA aggiunge un secondo metodo di autenticazione per evitare che il pool di utenti faccia affidamento solo sul nome utente e la password. Come secondo fattore per l'accesso degli utenti puoi utilizzare messaggi SMS o password monouso basate sul tempo (TOTP). Puoi inoltre utilizzare l'autenticazione adattiva con il modello basato su rischio per prevedere quando potrebbe essere necessario un altro fattore di autenticazione. Le funzionalità di sicurezza avanzata del pool di utenti includono l'autenticazione adattiva e la protezione contro le credenziali compromesse.

Argomenti

- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Aggiunta di sicurezza avanzata a un bacino d'utenza
- Associazione di un ACL Web a un pool di utenti AWS WAF
- Distinzione tra maiuscole e minuscole del bacino d'utenza
- Protezione da eliminazione del bacino d'utenza
- Gestione delle risposte agli errori relativi all'esistenza degli utenti

Aggiunta dell'autenticazione MFA a un bacino d'utenza

L'autenticazione a più fattori (MFA) aumenta la sicurezza dell'app. Aggiunge un fattore di autenticazione elementi che possiedi al fattore elementi che conosci di nome utente e password. Come secondo fattore per l'accesso degli utenti puoi scegliere di utilizzare messaggi SMS o password monouso a tempo (TOTP).

> **Note**
>
> La prima volta che un nuovo utente accede all'app, Amazon Cognito emette i token OAuth 2.0 anche se il bacino d'utenza richiede MFA. Il secondo fattore di autenticazione quando l'utente accede per la prima volta è la conferma del messaggio di verifica inviata da Amazon Cognito. Se il bacino d'utenza richiede MFA, Amazon Cognito richiede all'utente di registrare un ulteriore fattore di accesso da utilizzare per ogni tentativo di accesso dopo il primo.

Con l'autenticazione adattiva, puoi configurare il bacino d'utenza in modo da richiedere l'autenticazione tramite secondo fattore in risposta all'aumento del livello di rischio. Per aggiungere l'autenticazione adattiva al bacino d'utenza, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

Quando imposti l'autenticazione MFA su `required` per un bacino d'utenza, tutti gli utenti devono completare l'MFA per accedere. Per accedere, ogni utente deve impostare almeno un fattore MFA, ad esempio la configurazione di SMS o TOTP. Quando si imposta la MFA su `required`, è necessario includere la configurazione MFA nell'onboarding degli utenti in modo che il bacino d'utenza consenta loro di accedere.

Se abiliti gli SMS come fattore MFA, puoi richiedere che gli utenti forniscano il loro numero di telefono e che lo verifichino durante la registrazione. Se l'autenticazione MFA è impostata su `required` e supporta solo SMS come un fattore, gli utenti devono fornire i numeri di telefono. Gli utenti senza numeri di telefono avranno bisogno del tuo supporto per aggiungere un numero di telefono al proprio profilo prima di poter accedere. È possibile utilizzare numeri di telefono non verificati quando gli SMS sono un fattore dell'MFA. Questi numeri riceveranno lo stato di verificato dopo l'esito positivo dell'MFA.

Se hai impostato MFA come obbligatoria e hai attivato SMS e TOTP come metodi di verifica supportati, Amazon Cognito chiede ai nuovi utenti senza numeri di telefono di configurare TOTP come fattore MFA. Se hai impostato MFA come obbligatoria e l'unico metodo MFA che hai attivato

è TOTP, Amazon Cognito chiede a tutti i nuovi utenti di configurare TOTP come fattore MFA la seconda volta che effettuano l'accesso. Amazon Cognito genera una sfida per configurare TOTP MFA in risposta InitiateAutha operazioni API. AdminInitiateAuth

L'interfaccia utente ospitata richiede agli utenti di configurare MFA quando vien impostata come obbligatoria. Quando imposti MFA come facoltativa nel pool di utenti, l'interfaccia utente ospitata non invia richieste agli utenti. Per utilizzare MFA opzionale, devi creare un'interfaccia nell'app che richiede agli utenti di scegliere se desiderano configurare MFA, quindi devi guidarli attraverso gli input dell'API per verificare il fattore di accesso aggiuntivo.

Dopo cinque tentativi non riusciti di presentazione di un codice MFA, Amazon Cognito avvia il processo di blocco con timeout esponenziale descritto in Flusso di autenticazione del bacino d'utenza.

Argomenti

- Prerequisiti
- Configurazione dell'autenticazione a più fattori
- MFA con SMS
- Autenticazione MFA con token di software TOTP

Prerequisiti

Prima di configurare l'MFA, valuta quanto segue:

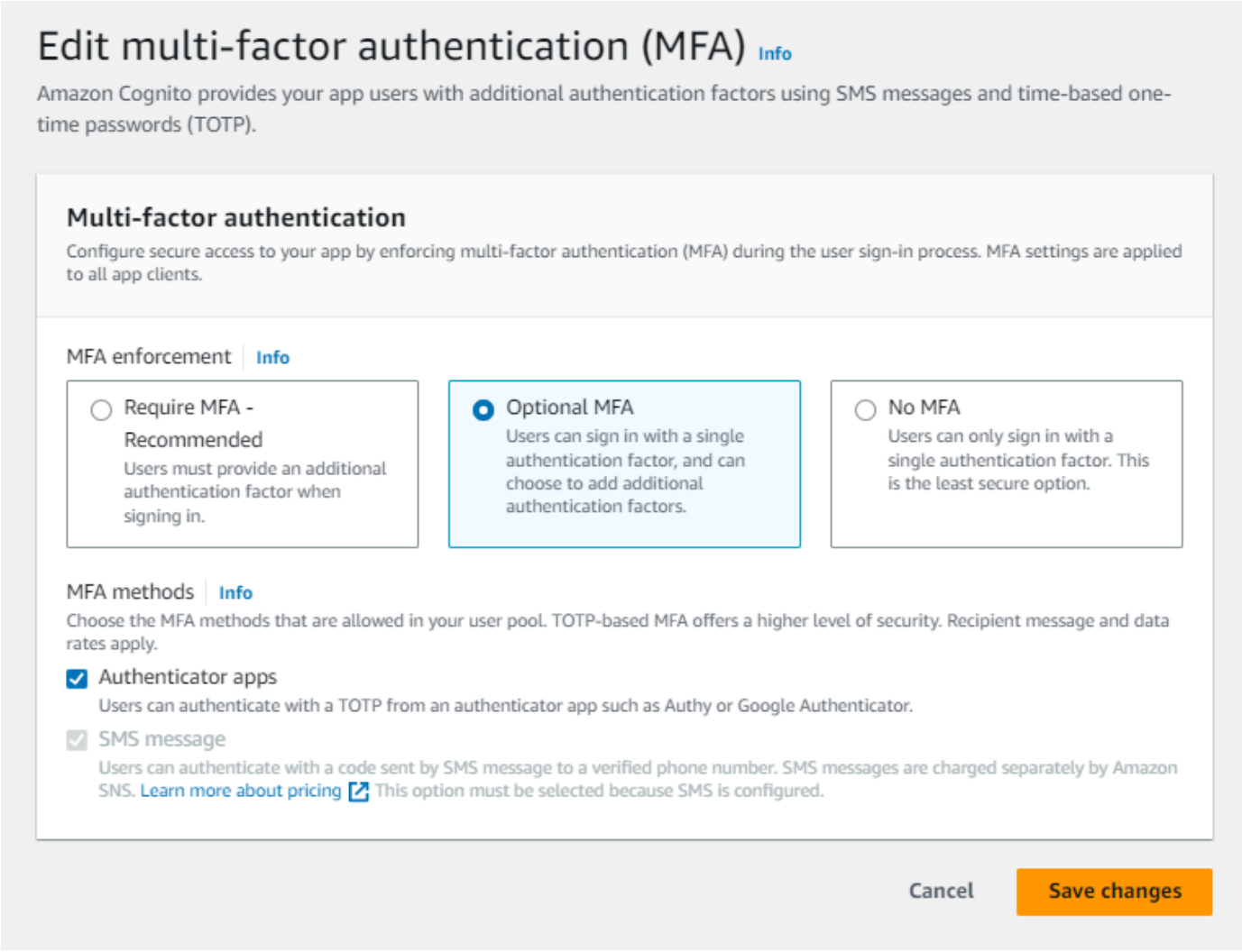
- Quando attivi l'MFA nel bacino d'utenza e scegli SMS come secondo fattore, puoi inviare messaggi SMS a un attributo numero di telefono che non hai verificato in Amazon Cognito. Dopo che il tuo utente ha completato l'MFA tramite SMS, Amazon Cognito imposta il suo `phone_number_verified`attributo su `true`.
- Se il tuo account si trova nella sandbox SMS Regione AWS che contiene le risorse Amazon Simple Notification Service (Amazon SNS) per il tuo pool di utenti, devi verificare i numeri di telefono in Amazon SNS prima di poter inviare un messaggio SMS. Per ulteriori informazioni, consulta Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito.
- Le caratteristiche di sicurezza avanzata richiedono che l'autenticazione MFA sia abilitata e impostata come facoltativa nella console del bacino d'utenza di Amazon Cognito. Per ulteriori informazioni, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

Configurazione dell'autenticazione a più fattori

Puoi configurare MFA nella console Amazon Cognito.

Come configurare l'autenticazione MFA nella console Amazon Cognito

1. Accedi alla console Amazon Cognito.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Sign-in experience (Esperienza di accesso). Individua Multi-factor authentication (Autenticazione a più fattori) e scegli Edit (Modifica)
5. Scegli il metodo di applicazione dell'MFA che desideri utilizzare con il bacino d'utenza.



   a. Richiedi MFA. Tutti gli utenti del bacino d'utenza devono accedere con un codice SMS aggiuntivo o con una password monouso a tempo (TOTP).

b. MFA facoltativa. È possibile offrire agli utenti la possibilità di registrare un ulteriore fattore di accesso e consentire comunque l'accesso agli utenti che non hanno configurato l'MFA. Scegli questa opzione se utilizzi l'autenticazione adattiva. Per ulteriori informazioni sull'autenticazione adattiva, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

c. Nessuna MFA. Gli utenti non possono registrare un ulteriore fattore di accesso.

6. Scegli i metodi MFA da supportare nell'app. È possibile impostare SMS o app di autenticazione che generano TOTP come secondo fattore. Si consiglia di implementare una MFA basata su TOTP in modo da poter utilizzare i messaggi SMS per il recupero dell'account.
7. Se utilizzi gli SMS come secondo fattore e non disponi di un ruolo IAM configurato per l'utilizzo con Amazon Simple Notification Service (Amazon SNS) per gli SMS, puoi crearne uno nella console. Nella scheda Messaging (Messaggistica) per il bacino d'utenza, individua SMS e scegli Edit (Modifica). Puoi anche utilizzare un ruolo esistente che consente ad Amazon Cognito di inviare messaggi SMS agli utenti per tuo conto. Per ulteriori informazioni, consulta IAM Roles (Ruoli IAM).
8. Scegli Save changes (Salva modifiche).

MFA con SMS

Quando un utente accede con l'autenticazione MFA abilitata, per prima cosa inserisce e invia il nome utente e la password. L'app client riceve una risposta `getMFA` che indica dove è stato inviato il codice di autorizzazione. L'app client dovrebbe indicare all'utente dove cercare il codice (ad esempio, a quale numero di telefono è stato inviato il codice). Successivamente, fornisce un modulo per l'immissione del codice. Infine, l'app client invia il codice per completare il processo di accesso. La destinazione è mascherata, ovvero tutte le cifre del numero di telefono tranne le ultime quattro sono nascoste. Se un'app sta utilizzando l'interfaccia utente ospitata di Amazon Cognito, viene visualizzata una pagina che consente all'utente di inserire il codice MFA.

Il codice di autorizzazione ricevuto mediante SMS è valido per la durata definita nell'opzione Authentication flow session duration (Durata della sessione del flusso di autenticazione) impostata per il client dell'app.

Imposta la durata di una sessione del flusso di autenticazione nella console Amazon Cognito nella scheda App integration (Integrazione app), quando modifichi il client dell'app in App clients and analytics (Client di app e analisi dei dati). È possibile impostare la durata della sessione del flusso di autenticazione anche all'interno della richiesta API `CreateUserPoolClient` o

`UpdateUserPoolClient`. Per ulteriori informazioni, consulta Flusso di autenticazione del bacino d'utenza.

Se un utente non dispone più dell'accesso al dispositivo a cui sono stati inviati via SMS i codici MFA, deve richiedere supporto al tuo ufficio di assistenza clienti. Un amministratore con Account AWS le autorizzazioni necessarie può modificare il numero di telefono dell'utente, ma solo tramite l' AWS CLI API o.

Quando un utente procede correttamente nell'autenticazione MFA con SMS, il suo numero di telefono viene contrassegnato come verificato.

> **Note**
>
> I costi degli SMS per l'autenticazione MFA sono addebitati separatamente. (Per l'invio di codici di verifica via e-mail, invece, non viene addebitato alcun costo). Per le informazioni sui prezzi di Amazon SNS, consulta la pagina Prezzi Worldwide SMS. Per la lista dei paesi in cui è disponibile la messaggistica SMS, consulta la pagina sulle regioni e i paesi supportati.

> **Important**
>
> Per garantire l'invio di SMS per verificare i numeri di telefono e per l'autenticazione MFA con SMS, è necessario richiedere ad Amazon SNS un aumento del limite di spesa.
> Amazon Cognito utilizza Amazon SNS per inviare SMS agli utenti. Il numero di messaggi SMS che Amazon SNS invia è soggetto a limiti di spesa. I limiti di spesa possono essere specificati per un AWS account e per singoli messaggi e si applicano solo al costo di invio di messaggi SMS.
> Di default, il limite di spesa per account (se non diversamente specificato) è pari a 1,00 USD al mese. Se desideri aumentare il limite, invia una richiesta di aumento del limite SNS al AWS Support Centro. In New limit value (Nuovo valore limite), immetti il limite di spesa mensile desiderato. Nel campo Use Case Description (Descrizione caso d'uso), indica che stai richiedendo un aumento del limite di spesa mensile per gli SMS.

Per aggiungere l'autenticazione MFA al bacino d'utenza, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza. Per ulteriori informazioni sui messaggi SMS con Amazon SNS nel tuo pool di utenti, consulta. Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito

Autenticazione MFA con token di software TOTP

Quando configuri l'autenticazione MFA con token di software TOTP nel pool di utenti, l'utente accede con nome utente e password, quindi utilizza un TOTP per completare l'autenticazione. Dopo che l'utente imposta e verifica nome utente e password, può attivare un token software TOTP per l'autenticazione MFA. Se l'app utilizza l'interfaccia utente ospitata da Amazon Cognito per consentire l'accesso degli utenti, l'utente invia il nome utente e la password, quindi invia la password TOTP in un pagina di accesso aggiuntiva.

Puoi attivare l'autenticazione MFA con TOTP per il bacino d'utenza nella console Amazon Cognito oppure utilizzando le operazioni API di Amazon Cognito. A livello di pool di utenti, puoi chiamare SetUserPoolMfaConfigper configurare l'MFA e abilitare TOTP MFA.

> **Note**
>
> Se l'autenticazione MFA con token software TOTP non è attivata per il bacino d'utenza, Amazon Cognito non può usare i token per associare o verificare gli utenti. In questo caso, gli utenti ricevono un'eccezione `SoftwareTokenMFANotFoundException` con la descrizione `Software Token MFA has not been enabled by the userPool`. Se in seguito disattivi l'autenticazione MFA tramite token software per il bacino d'utenza, gli utenti che hanno precedentemente associato e verificato un token TOTP possono continuare a utilizzarlo per l'MFA.

La configurazione del metodo TOTP per gli utenti è un processo in più fasi in cui l'utente riceve un codice segreto che convalida immettendo una password monouso. Quindi, puoi abilitare l'MFA con TOTP per l'utente o impostare la password TOTP come metodo preferito di MFA per l'utente.

Quando configuri il tuo pool di utenti in modo che venga richiesta l'autenticazione MFA con token di software TOTP e i tuoi utenti si iscrivono alla tua app nell'interfaccia utente ospitata, Amazon Cognito automatizza il processo utente. Amazon Cognito richiede all'utente di scegliere un metodo di autenticazione MFA, visualizza un codice QR per configurare l'app Authenticator e verifica la registrazione MFA. Nei pool di utenti in cui si è consentito agli utenti di scegliere tra l'autenticazione MFA con SMS e con token di software TOTP, Amazon Cognito consente all'utente di scegliere uno dei metodi disponibili. Per ulteriori informazioni sull'esperienza di registrazione all'interfaccia utente ospitata, consulta Informazioni su come registrare un nuovo account nell'interfaccia utente ospitata di Amazon Cognito.

> **Important**
>
> Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP dell'interfaccia utente ospitata. Per creare una regola contenente un'azione CAPTCHA, ma che non abbia alcun effetto sul token di software TOTP dell'interfaccia utente ospitata, consulta Configurazione dell'ACL AWS WAF Web per l'interfaccia utente TOTP MFA ospitata. Per ulteriori informazioni sugli ACL AWS WAF Web e Amazon Cognito, consulta. Associazione di un ACL Web a un pool di utenti AWS WAF

Per implementare l'autenticazione MFA con token di software TOTP in un'interfaccia utente personalizzata in cui si utilizza l'API Amazon Cognito, consulta Configurazione dell'autenticazione MFA per un utente nell'API dei pool di utenti Amazon Cognito.

Per aggiungere l'autenticazione MFA al bacino d'utenza, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.

Considerazioni e limitazioni riguardo all'MFA con TOTP

1. Amazon Cognito supporta l'MFA con token software tramite un'app di autenticazione che genera i codici TOTP. Amazon Cognito non supporta l'MFA basata su hardware.
2. Quando il tuo bacino d'utenza richiede una TOTP per un utente che non l'ha configurata, l'utente riceve un token di accesso monuso che la tua app può utilizzare per attivare l'MFA con TOTP per l'utente. I tentativi di accesso successivi non riescono finché l'utente non registra un ulteriore fattore di accesso con TOTP.
   - Un utente che si iscrive al tuo bacino d'utenza con l'operazione API `SignUp` o tramite l'interfaccia utente ospitata, riceve token monouso una volta che ha completato la registrazione.
   - Una volta che hai creato un utente e che l'utente ha impostato la password iniziale, Amazon Cognito emette token monouso per l'utente dall'interfaccia utente ospitata. Se imposti una password permanente per l'utente, Amazon Cognito emette token monouso quando l'utente accede per la prima volta.
   - Amazon Cognito non rilascia token monouso a un utente creato dall'amministratore che accede con le operazioni o API. InitiateAuthAdminInitiateAuth Dopo che l'utente ha impostato correttamente la password iniziale o che tu hai impostato una password permanente per l'utente, Amazon Cognito richiede immediatamente all'utente di configurare l'MFA.

3. Se un utente in un bacino d'utenza che richiede l'MFA ha già ricevuto un token di accesso monouso ma non ha configurato l'TOTP con MFA, l'utente non può accedere con l'interfaccia utente ospitata fino a quando non ha configurato l'MFA. Invece del token di accesso, puoi utilizzare il valore di `session` risposta di una `MFA_SETUP` sfida o contenuto in una richiesta. InitiateAuthAdminInitiateAuthAssociateSoftwareToken
4. Se gli utenti hanno impostato la TOTP, possono utilizzarla per l'MFA anche se in seguito la disattivi per il bacino d'utenza.
5. Amazon Cognito accetta TOTP solo dalle app di autenticazione che generano codici con la funzione hash SHA-1. I codici generati con l'hashing SHA-256 restituiscono un errore `Code mismatch`.

Configurazione dell'autenticazione MFA per un utente nell'API dei pool di utenti Amazon Cognito

Quando un utente accede per la prima volta, l'app utilizza il token di accesso monouso per generare la chiave privata TOTP e presentarla all'utente in formato testo o codice QR. L'utente configura la propria app di autenticazione e fornisce un TOTP per i successivi tentativi di accesso. La tua app o l'interfaccia utente ospitata presenta la TOTP ad Amazon Cognito nelle risposte alle sfide MFA.

Argomenti

- Associazione del token di software TOTP
- Verifica del token TOTP
- Accesso usando l'MFA con TOTP
- Rimozione del token TOTP

Associazione del token di software TOTP

Per associare il token TOTP, invia al tuo utente un codice segreto che deve convalidare con una password monouso. Per l'associazione del token sono necessari tre passaggi.

1. Quando l'utente sceglie il token software TOTP MFA, chiama AssociateSoftwareTokenper restituire un codice chiave segreto condiviso generato univoco per l'account utente. Puoi autorizzare AssociateSoftwareToken con un token di accesso o una stringa di sessione.
2. La tua app presenta all'utente la chiave privata o un codice QR generato dalla chiave privata. L'utente deve inserire la chiave in un'app per la generazione di password TOTP come Google Authenticator. È possibile utilizzare libqrencode per generare un codice QR.

3. L'utente inserisce la chiave o scansiona il codice QR in un'app di autenticazione come Google Authenticator che inizia a generare codici.

Verifica del token TOTP

Poi, verifica del token TOTP. Richiedi codici di esempio dal tuo utente e forniscili al servizio Amazon Cognito per confermare che l'utente sta generando correttamente i codici TOTP, come indicato di seguito.

1. L'app richiede all'utente un codice per dimostrare di aver configurato correttamente l'app di autenticazione.
2. L'app di autenticazione dell'utente mostra una password temporanea. L'app di autenticazione crea la password in base alla chiave segreta che hai fornito all'utente.
3. L'utente inserisce la password temporanea. La tua app passa la password temporanea ad Amazon Cognito in una Richiesta API `VerifySoftwareToken`.
4. Amazon Cognito ha mantenuto la chiave segreta associata all'utente, genera una TOTP e la confronta con quella fornita dall'utente. Se corrispondono,`VerifySoftwareToken` restituisce una risposta `SUCCESS`.
5. Amazon Cognito associa il fattore TOTP all'utente.
6. Se l'operazione `VerifySoftwareToken` restituisce una risposta `ERROR`, assicurati che il segnale di clock dell'utente sia corretto e che l'utente non abbia superato il numero massimo di tentativi. Amazon Cognito accetta token TOTP entro 30 secondi prima o dopo il tentativo, per tenere conto del minore sfasamento del segnal di clock. Una volta risolto il problema, riprova a VerifySoftwareToken eseguire l'operazione.

Accesso usando l'MFA con TOTP

A questo punto, l'utente accede con la password una tantum a tempo. Il procedimento è riportato di seguito.

1. L'utente inserisce il nome utente e la password per accedere all'app client.
2. Viene invocata la richiesta dell'MFA con TOTP e l'app chiede all'utente di immettere una password temporanea.
3. L'utente ottiene la password temporanea da un'app per la generazione di password TOTP associata.

4. L'utente inserisce il codice TOTP nell'app client. L'app notifica al servizio Amazon Cognito di verificarlo. Per ogni accesso, RespondToAuthChallengedeve essere chiamato per ottenere una risposta alla nuova sfida di autenticazione TOTP.
5. Se il token è verificato da Amazon Cognito, l'accesso viene completato correttamente e l'utente procede nel flusso di autenticazione.

Rimozione del token TOTP

Infine, l'app dovrebbe consentire all'utente di disattivare la configurazione TOTP. Al momento non puoi eliminare il token di software TOTP di un utente. Per sostituire il token di software dell'utente, associa e verifica un nuovo token di software. Per disattivare TOTP MFA per un utente, chiama SetUserMFAPREference per modificare l'utente in modo che non utilizzi MFA o solo SMS MFA.

1. Crea un'interfaccia nell'app per gli utenti che desiderano reimpostare MFA. Richiedi a un utente in questa interfaccia di inserire la password.
2. Se Amazon Cognito restituisce una sfida TOTP MFA, aggiorna la preferenza MFA dell'utente con MFAReference. SetUser
3. Nell'app, comunica all'utente che MFA è stata disattivata e chiedi di effettuare nuovamente l'accesso.

Configurazione dell'ACL AWS WAF Web per l'interfaccia utente TOTP MFA ospitata

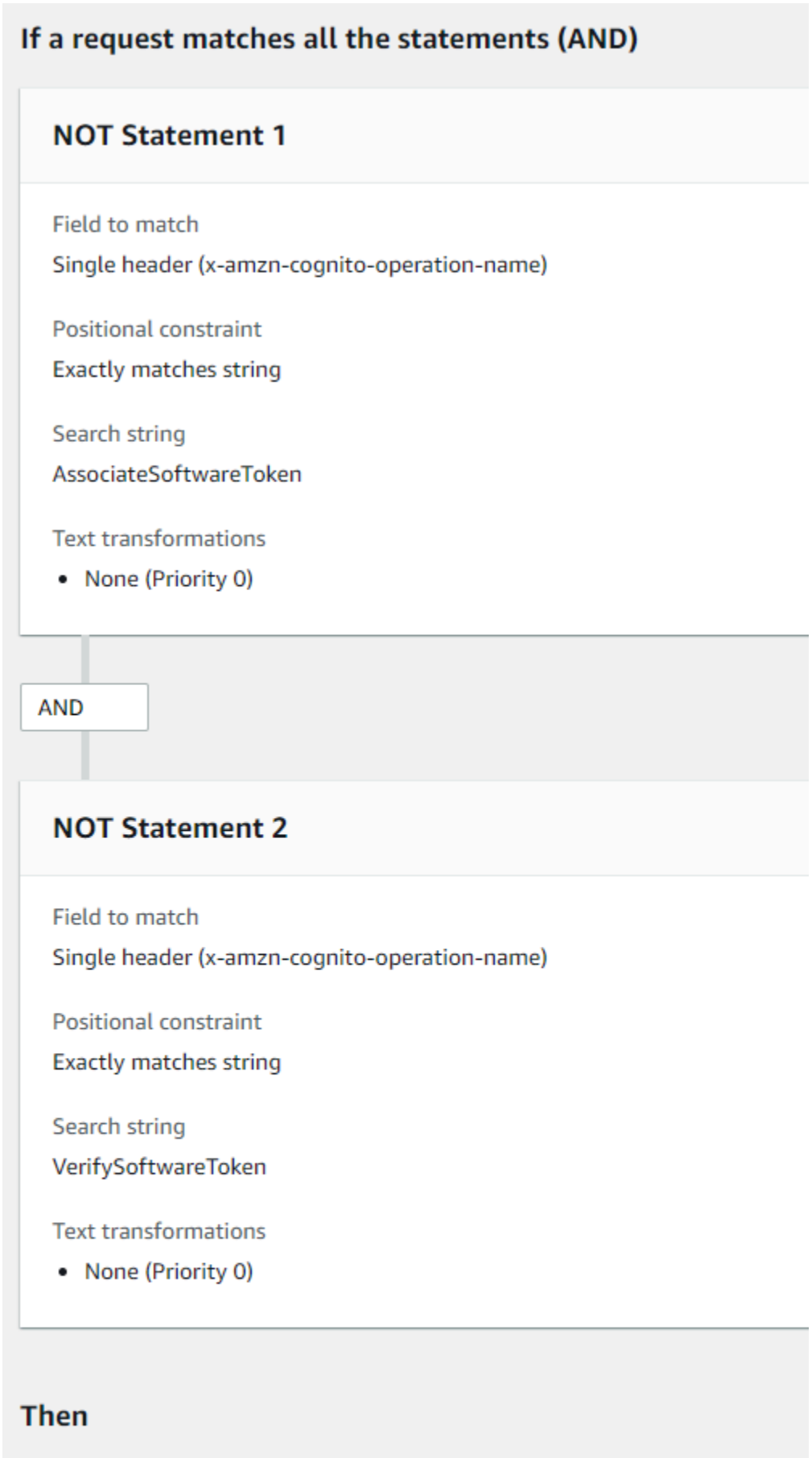
Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP dell'interfaccia utente ospitata. AWS WAF Le regole CAPTCHA influiscono solo sull'MFA TOTP nell'interfaccia utente ospitata in questo modo. L'autenticazione MFA con SMS rimane invariata.

Amazon Cognito visualizza il seguente errore quando la regola CAPTCHA non consente a un utente di completare la configurazione dell'autenticazione MFA con token di software TOTP.

Request not allowed due to WAF captcha (Richiesta non consentita a causa di un'azione CAPTCHA WAF).

Questo errore si verifica quando viene AWS WAF richiesto un CAPTCHA in risposta alle AssociateSoftwareTokenrichieste VerifySoftwareTokenAPI effettuate in background dal pool di utenti. Per creare una regola che preveda un'azione CAPTCHA e non influisca sul token di software TOTP dell'interfaccia utente ospitata, escludi i valori di intestazione `x-amzn-cognito-operation-name` di `AssociateSoftwareToken` e `VerifySoftwareToken` dall'azione CAPTCHA nella regola.

La schermata seguente mostra una AWS WAF regola di esempio che applica un'azione CAPTCHA a tutte le richieste che non hanno un valore di intestazione pari a o. `x-amzn-cognito-operation-name` `AssociateSoftwareToken` `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.

Per ulteriori informazioni sugli ACL AWS WAF Web e Amazon Cognito, consulta. Associazione di un ACL Web a un pool di utenti AWS WAF

Aggiunta di sicurezza avanzata a un bacino d'utenza

Dopo aver creato il bacino d'utenza, potrai accedere ad Advanced security (Sicurezza avanzata) sulla barra di navigazione della console Amazon Cognito. Puoi attivare le caratteristiche di sicurezza avanzata per il bacino d'utenza e personalizzare le azioni che vengono eseguite in risposta a rischi differenti. In alternativa, si può utilizzare la modalità di controllo per raccogliere i parametri sui rischi rilevati, senza applicare mitigazioni della sicurezza. In modalità di controllo, le funzionalità di sicurezza avanzate pubblicano i parametri su Amazon CloudWatch. Puoi visualizzare le metriche di sicurezza avanzate dopo che Amazon Cognito ha generato il suo primo evento di sicurezza avanzato. Per informazioni, consulta Visualizzazione dei parametri di sicurezza avanzata.

Le funzionalità di sicurezza avanzate includono rilevamento delle credenziali compromesse e autenticazione adattiva.

Credenziali compromesse

Gli utenti riutilizzano le password per più account utente. La funzionalità credenziali compromesse di Amazon Cognito compila i dati di fughe pubbliche di nomi utente e password e confronta le credenziali degli utenti con gli elenchi di credenziali divulgate. Il rilevamento delle credenziali compromesse verifica inoltre la presenza di password facilmente indovinate.

Puoi scegliere le azioni dell'utente che richiedono la verifica della presenza di credenziali compromesse e l'azione che deve effettuare Amazon Cognito in risposta. Per gli eventi di accesso, registrazione e modifica della password, Amazon Cognito può scegliere tra le opzioni Blocca l'accesso o Consenti l'accesso. In entrambi i casi, Amazon Cognito genera un log delle attività degli utenti, in cui è possibile trovare ulteriori informazioni sull'evento.

Autenticazione adattiva

Amazon Cognito può esaminare le informazioni sulla posizione e il dispositivo delle richieste di accesso degli utenti e applicare una risposta automatica per proteggere gli account utente del pool di utenti da attività sospette.

Quando attivi la sicurezza avanzata, Amazon Cognito assegna un punteggio di rischio all'attività dell'utente. Puoi assegnare una risposta automatica all'attività sospetta: puoi scegliere tra le opzioni Richiedi MFA, Blocca l'accesso o semplicemente registrare i dettagli dell'attività e il punteggio di rischio. Puoi anche inviare automaticamente messaggi e-mail che informano

l'utente dell'attività sospetta in modo che possa reimpostare la password o eseguire altre azioni autogestite.

Personalizzazione del token di accesso

Quando attivi le funzionalità di sicurezza avanzate, puoi configurare il pool di utenti per accettare le risposte a un evento trigger Lambda versione 2. Con la versione 2, puoi personalizzare gli ambiti e altre attestazioni nei token di accesso. Ciò aumenta la capacità di creare risultati di autorizzazione flessibili quando gli utenti eseguono l'autenticazione. Per ulteriori informazioni, consulta Personalizzazione del token di accesso.

Argomenti

- Considerazioni e limitazioni
- Prerequisiti
- Configurazione delle caratteristiche di sicurezza avanzata
- Verifica della presenza di credenziali compromesse
- Utilizzo dell'autenticazione adattiva
- Visualizzazione dei parametri di sicurezza avanzata
- Attivazione della sicurezza avanzata del bacino d'utenza dall'app

Considerazioni e limitazioni

- Per le funzionalità di sicurezza avanzate di Amazon Cognito si applicano dei costi aggiuntivi. Consulta la pagina dei prezzi di Amazon Cognito.
- Amazon Cognito supporta l'autenticazione adattiva e il rilevamento delle credenziali compromesse con i seguenti flussi di autenticazione standard:, e. `USER_PASSWORD_AUTH` `ADMIN_USER_PASSWORD_AUTH` `USER_SRP_AUTH` Non si può utilizzare la sicurezza avanzata con un flusso `CUSTOM_AUTH` e Trigger Lambda di richieste di autenticazione personalizzate, o con accesso federato.
- Grazie alle funzionalità di sicurezza avanzate di Amazon Cognito nella modalità Funzione completa, è possibile creare eccezioni Blocca sempre e Consenti sempre dell'indirizzo IP. A una sessione il cui indirizzo IP è nella lista delle eccezioni Blocca sempre non viene assegnato un livello di rischio tramite autenticazione adattiva e non può accedere al bacino d'utenza.
- Le richieste bloccate provenienti da indirizzi IP nella lista delle eccezioni Blocca sempre nel tuo baclno d'utenza, contribuiscono alle quote tariffarie per le richieste dei tuoi bacini d'utenza. Le

funzionalità di sicurezza avanzate di Amazon Cognito non sono efficaci contro gli attacchi DDoS (Distributed Denial of Service). Per implementare difese contro gli attacchi volumetrici nei tuoi pool di utenti, aggiungi ACL web. AWS WAF Per ulteriori informazioni, consulta Associazione di un ACL Web a un pool di utenti AWS WAF.

- Le credenziali concesse ai client sono destinate all'autorizzazione machine-to-machine (M2M) senza connessione agli account utente. Le funzionalità di sicurezza avanzata monitorano solo gli account utente e le password nel pool di utenti. Per implementare funzionalità di sicurezza nella tua attività M2M, prendi in considerazione le funzionalità di monitoraggio della frequenza delle AWS WAF richieste e dei contenuti. Per ulteriori informazioni, consulta Associazione di un ACL Web a un pool di utenti AWS WAF.

Prerequisiti

Prima di iniziare, avrai bisogno di:

- Un bacino d'utenza con un client dell'app. Per ulteriori informazioni, consulta Nozioni di base sui bacini d'utenza.
- Imposta l'autenticazione a più fattori (MFA) su Facoltativa nella console Amazon Cognito per utilizzare la caratteristica di autenticazione adattiva basata sul rischio. Per ulteriori informazioni, consulta Aggiunta dell'autenticazione MFA a un bacino d'utenza.
- Se utilizzi l'indirizzo e-mail per le notifiche, passa alla console Amazon SES per configurare e verificare un indirizzo e-mail o un dominio da utilizzare con le notifiche e-mail. Per ulteriori informazioni su Amazon SES, consulta Verifica delle identità in Amazon SES.

Configurazione delle caratteristiche di sicurezza avanzata

Puoi configurare le caratteristiche di sicurezza avanzate di Amazon Cognito nella AWS Management Console.

Per configurare la sicurezza avanzata per un bacino d'utenza

1. Passa alla console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.

4. Scegli la scheda App integration (Integrazione app). Individua Advanced security (Sicurezza avanzata) e scegli Enable (Abilita). Se hai abilitato la sicurezza avanzata in precedenza, scegli Edit (Modifica).
5. Seleziona Full function (Funzione completa) per configurare le risposte di sicurezza avanzate alle credenziali compromesse e all'autenticazione adattiva. Seleziona Controlla solo per raccogliere informazioni e inviare i dati del pool di utenti a. CloudWatch Il prezzo di sicurezza avanzata si applica sia alla modalità Audit only (Solo audit) sia a quella Full function (Funzione completa). Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

   Consigliamo di tenere le funzionalità di sicurezza avanzata in modalità di controllo per due settimane prima di abilitare le operazioni. In questo periodo, Amazon Cognito apprende i modelli di utilizzo degli utenti dell'app.
6. Se hai selezionato Audit only (Solo audit), scegli Save changes (Salva modifiche). Se hai selezionato Full function (Funzione completa):

   a. Seleziona se intendi intraprendere operazioni personalizzate o utilizzare le impostazioni predefinite di Cognito in risposta al sospetto di credenziali compromesse. Le Impostazioni predefinite di Cognito sono:

      i. Rilevamento di credenziali compromesse all'accesso, alla registrazione e in caso di modifica della password.

      ii. Rispondi alla compromissione delle credenziali con l'operazione Blocca accesso.

   b. Se hai selezionato Custom (Personalizza) per le operazioni relative alle Compromised credentials (Credenziali compromesse), scegli le operazioni del bacino d'utenza che Amazon Cognito utilizzerà per il Rilevamento di eventi e le Risposte alle credenziali compromesse che vorresti che Amazon Cognito mettesse in atto. Puoi eseguire le operazioni Block sign-in (Blocca accesso) o Allow sign-in (Consenti accesso) nel caso in cui vi sia il sospetto che le credenziali siano state compromesse.

   c. Scegli come rispondere ai tentativi di accesso dannosi alla voce Adaptive authentication (Autenticazione adattiva). Seleziona se intendi intraprendere operazioni personalizzate o utilizzare le impostazioni predefinite di Cognito in risposta a sospette attività dannose. Selezionando le Impostazioni di default di Cognito, Amazon Cognito bloccherà l'accesso a tutti i livelli di rischio e non notificherà l'utente.

   d. Se hai selezionato Custom (Personalizza) per Adaptive authentication (Autenticazione adattiva), scegli le operazioni di Automatic risk response (Risposta automatica al rischio) che Amazon Cognito intraprenderà in risposta ai rischi rilevati in base al livello di gravità.

Quando si assegna una risposta a un livello di rischio, non è possibile assegnare una risposta meno restrittiva a un livello di rischio più elevato. È possibile assegnare le seguenti risposte ai livelli di rischio:

i. Allow sign-in (Consenti l'accesso): non intraprendere alcuna operazione preventiva.

ii. Optional MFA (MFA facoltativo): se l'utente ha configurato l'MFA, Amazon Cognito gli richiederà sempre di fornire un ulteriore fattore SMS o una password a tempo (TOTP) quando effettua l'accesso. Se l'utente non ha configurato MFA, può continuare ad accedere normalmente.

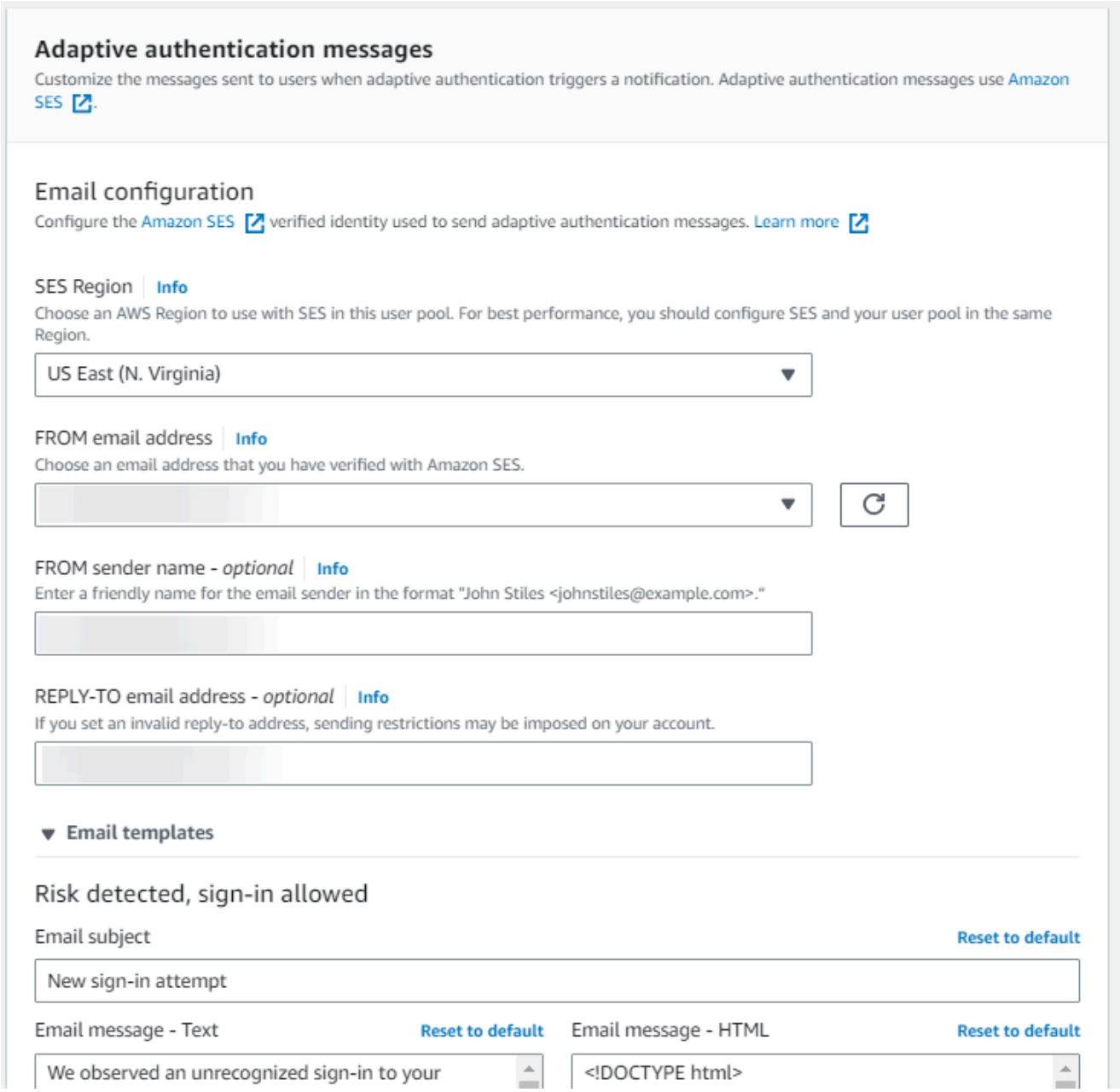
iii. Require MFA (MFA obbligatorio): se l'utente ha configurato MFA, Amazon Cognito gli richiederà sempre di fornire un fattore SMS aggiuntivo o un fattore TOTP al momento dell'accesso. Se l'utente non ha l'MFA configurato, Amazon Cognito richiederà di configurare l'MFA. Prima di richiedere automaticamente l'MFA per i tuoi utenti, configura un sistema nell'app che consenta di acquisire i numeri di telefono per l'MFA tramite SMS o di registrare app di autenticazione per l'MFA tramite TOTP.

iv. Block sign-in (Blocca l'accesso): impedisci all'utente di accedere.

v. Notify user (Notifica l'utente): invia un'e-mail all'utente con informazioni sul rischio rilevato da Amazon Cognito e sulla risposta che hai messo in atto. È possibile personalizzare i modelli di messaggio e-mail per i messaggi inviati.

7. Se hai scelto Notify user (Notifica l'utente) nel passaggio precedente, puoi personalizzare le impostazioni di recapito e i modelli di messaggio e-mail per l'autenticazione adattiva.

   a. Alla voce Email configuration (Configurazione e-mail), scegli la Regione SES, l'Indirizzo e-mail FROM, il Nome del mittente FROM e l'indirizzo e-mail REPLY-TO da utilizzare con l'autenticazione adattiva. Per ulteriori informazioni sull'integrazione di messaggi e-mail per il bacino d'utenza con Amazon Simple Email Service, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito.



b. Espandi Email templates (Modelli e-mail) per personalizzare le notifiche di autenticazione adattiva per i messaggi e-mail in versione HTML e in testo normale. Per ulteriori informazioni sui modelli di messaggio e-mail, consulta Modelli dei messaggi.

8. Espandi IP address exceptions (Eccezioni agli indirizzi IP) per creare un elenco di indirizzi IPv4 o IPv6 da consentire sempre o bloccare sempre che saranno sempre consentiti o bloccati, indipendentemente dalla valutazione avanzata del rischio per la sicurezza. Specifica gli intervalli di indirizzi IP nella notazione CIDR (ad esempio, 192.168.100.0/24).

9. Scegli Save changes (Salva modifiche).

Verifica della presenza di credenziali compromesse

Amazon Cognito è in grado di rilevare se il nome utente e la password di un utente sono stati compromessi. Questo può accadere quando gli utenti utilizzano le stesse credenziali su più di un sito oppure quando utilizzano password non sicure. Amazon Cognito verifica gli utenti locali che accedono con nome utente e password, nell'interfaccia utente ospitata e mediante l'API Amazon Cognito. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno.

In Advanced security (Sicurezza avanzata) nella scheda App integration (Integrazione app) della console Amazon Cognito, puoi configurare Compromised credentials (Credenziali compromesse). Configura Event detection (Rilevamento di eventi) per scegliere gli eventi utente da monitorare per individuare le credenziali compromesse. Configura Compromised credentials responses (Risposte delle credenziali compromesse) per scegliere se consentire o bloccare l'utente se vengono rilevate credenziali compromesse. Amazon Cognito può controllare la presenza di credenziali compromesse durante gli accessi, le registrazioni e le modifiche delle password.

Quando scegli Consenti l'accesso, puoi esaminare Amazon CloudWatch Logs per monitorare le valutazioni effettuate da Amazon Cognito sugli eventi degli utenti. Per ulteriori informazioni, consulta Visualizzazione dei parametri di sicurezza avanzata. Quando si sceglie Block sign-in (Blocca l'accesso), Amazon Cognito impedisce l'accesso da parte degli utenti che utilizzano credenziali compromesse. Quando Amazon Cognito blocca l'accesso a un utente, imposta il parametro `UserStatus` dell'utente su `RESET_REQUIRED`. Un utente con uno stato `RESET_REQUIRED` deve modificare la password prima di poter accedere di nuovo.

> **Note**
>
> Al momento, Amazon Cognito non verifica la presenza di credenziali compromesse per operazioni di accesso con flusso Secure Remote Password (SRP). SRP invia una prova di password con hash durante l'accesso. Amazon Cognito non dispone dell'accesso alle password internamente, quindi può solo valutare una password passata dal client in testo normale.
>
> Amazon Cognito verifica la presenza di credenziali compromesse negli accessi che utilizzano l'AdminInitiateAuthAPI con `ADMIN_USER_PASSWORD_AUTH` flow e l'InitiateAuthAPI con `USER_PASSWORD_AUTH` flow.

Per aggiungere misure di protezione contro le credenziali compromesse al bacino d'utenza, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

Utilizzo dell'autenticazione adattiva

Con l'autenticazione adattiva, puoi configurare il bacino d'utenza in modo da bloccare i tentativi di accesso sospetti o aggiungere l'autenticazione tramite secondo fattore in risposta all'aumento del livello di rischio. Per ogni tentativo di accesso, Amazon Cognito genera un punteggio di rischio che indica la probabilità che la richiesta di accesso provenga da un'origine compromessa. Il punteggio di rischio si basa su fattori che includono le informazioni sul dispositivo e sull'utente. L'autenticazione adattiva può attivare o richiedere l'autenticazione a più fattori (MFA) per un utente del pool di utenti quando Amazon Cognito rileva un rischio nella sessione di un utente che non ha ancora scelto un metodo MFA. Quando attivi l'autenticazione MFA, l'utente riceve sempre la richiesta di fornire o impostare un secondo fattore durante l'autenticazione, indipendentemente da come è stata configurata l'autenticazione adattiva. Dal punto di vista dell'utente, l'app offre le indicazioni per configurare l'autenticazione MFA e, facoltativamente, Amazon Cognito gli impedisce di accedere nuovamente fino a quando non configura un fattore aggiuntivo.

Amazon Cognito pubblica su Amazon i tentativi di accesso, i relativi livelli di rischio e le sfide non riuscite. CloudWatch Per ulteriori informazioni, consulta Visualizzazione dei parametri di sicurezza avanzata.

Per aggiungere l'autenticazione adattiva al bacino d'utenza, consulta Aggiunta di sicurezza avanzata a un bacino d'utenza.

Argomenti

- Panoramica dell'autenticazione adattiva
- Aggiunta di dati di dispositivo e sessione utente alle richieste API
- Visualizzazione della cronologia eventi dell'utente
- Fornitura del feedback sugli eventi
- Invio di messaggi di notifica

Panoramica dell'autenticazione adattiva

Nella scheda Integrazione app della pagina Sicurezza avanzata della console di Amazon Cognito puoi scegliere le impostazioni di autenticazione adattiva, incluse le azioni da eseguire in base ai diversi livelli di rischio e la personalizzazione dei messaggi di notifica per gli utenti. Puoi

assegnare una configurazione di sicurezza avanzata globale a tutti i client dell'app e applicare una configurazione a livello di client ai singoli client dell'app.

L'autenticazione adattiva di Amazon Cognito assegna uno dei seguenti livelli di rischio a ciascuna sessione utente: Alto, Medio, Basso o Nessun rischio.

Valuta con attenzione le opzioni disponibili quando modifichi il valore di Enforcement method (Metodo di applicazione) da Audit-only (Solo verifica) a Full-function (Funzione completa). Le risposte automatiche che applichi ai livelli di rischio influiscono sul livello di rischio che Amazon Cognito assegna alle sessioni utente successive che presentano le stesse caratteristiche. Ad esempio, dopo aver scelto di non intraprendere alcuna azione, o Allow (Consentire), le sessioni utente che Amazon Cognito inizialmente ha definito ad alto rischio, le successive sessioni simili vengono considerate da Amazon Cognito a basso rischio.

Per ogni livello di rischio, puoi scegliere tra le seguenti opzioni:

Opzione	Azione
Consenti	Gli utenti possono accedere senza un ulteriore fattore.
MFA facoltativa	Gli utenti per i quali è configurata l'autenti cazione con un secondo fattore, devono completare una seconda sfida per poter accedere. I secondi fattori disponibili sono la verifica tramite SMS e token di software TOTP. Gli utenti senza un secondo fattore configurato possono accedere con un solo set di credenzia li.
Richiedi MFA	Gli utenti per i quali è configurata l'autenti cazione con un secondo fattore, devono completare una seconda sfida per poter accedere. Amazon Cognito blocca l'accesso agli utenti che non hanno un secondo fattore configurato.
Blocco	Amazon Cognito blocca tutti i tentativi di accesso al livello di rischio designato.

> **Note**
>
> Non è necessario verificare i numeri di telefono da utilizzare per la ricezione di SMS come secondo fattore di autenticazione.

Aggiunta di dati di dispositivo e sessione utente alle richieste API

Puoi raccogliere e trasferire informazioni sulla sessione utente alla funzionalità di sicurezza avanzata di Amazon Cognito quando utilizzi l'API per effettuare la registrazione, l'accesso e la reimpostazione della password. Queste informazioni includono l'indirizzo IP dell'utente e un identificatore univoco del dispositivo.

Potrebbe essere presente un dispositivo di rete intermedio tra gli utenti e Amazon Cognito, ad esempio un servizio proxy o un server applicazioni. Puoi raccogliere i dati contestuali degli utenti e passarli ad Amazon Cognito in modo che l'autenticazione adattiva calcoli il rischio in base alle caratteristiche dell'endpoint utente, anziché al server o al proxy. Se l'app lato client chiama direttamente le operazioni dell'API Amazon Cognito, l'autenticazione adattiva registra automaticamente l'indirizzo IP di origine. Tuttavia, non registra altre informazioni sul dispositivo, ad esempio `user-agent`, a meno che non venga acquisita anche un'impronta del dispositivo.

Genera questi dati con la libreria di raccolta dati contestuali di Amazon Cognito e inviali alla sicurezza avanzata di Amazon Cognito con ContextDatai parametri and. UserContextData La libreria di raccolta dei dati contestuali è inclusa negli AWS SDK. Per maggiori informazioni, consulta Integrazione di Amazon Cognito con le app Web e per dispositivi mobili. È possibile passare `ContextData` se hai attivato le funzionalità di sicurezza avanzate nel pool di utenti. Per ulteriori informazioni, consulta Configurazione delle caratteristiche di sicurezza avanzata.

Quando si chiamano le seguenti operazioni API autenticate di Amazon Cognito dal server applicazioni, viene passato l'IP del dispositivo dell'utente nel parametro `ContextData`. Inoltre, passa il nome e il percorso del server, nonché i dati relativi all'impronta del dispositivo.

- AdminInitiateAuth
- AdminRespondToAuthChallenge

Quando chiami le operazioni API non autenticate di Amazon Cognito, puoi passare `UserContextData` alle funzionalità di sicurezza avanzata di Amazon Cognito. Questi dati includono

un'impronta del dispositivo nel parametro `EncodedData`. Puoi anche passare un parametro `IpAddress` in `UserContextData` se vengono soddisfatte le seguenti condizioni:

- Hai attivato le funzionalità di sicurezza avanzata nel pool di utenti. Per ulteriori informazioni, consulta Configurazione delle caratteristiche di sicurezza avanzata.
- Il client dell'app ha un segreto del client. Per ulteriori informazioni, consulta Configurazione di un client dell'app per un pool di utenti.
- Hai attivato l'opzione Accept additional user context data (Accetta dati utente contestuali aggiuntivi) nel client dell'app. Per ulteriori informazioni, consulta Accettazione di dati utente contestuali aggiuntivi (AWS Management Console).

L'app può popolare il parametro `UserContextData` con dati di impronte del dispositivo codificati e l'indirizzo IP del dispositivo dell'utente nelle seguenti operazioni API non autenticate di Amazon Cognito.

- InitiateAuth
- RespondToAuthChallenge
- SignUp
- ConfirmSignUp
- ForgotPassword
- ConfirmForgotPassword
- ResendConfirmationCode

Accettazione di dati utente contestuali aggiuntivi (AWS Management Console)

Il pool di utenti accetta un indirizzo IP nel parametro `UserContextData` dopo aver attivato la funzionalità Accept additional user context data (Accetta dati utente contestuali aggiuntivi). Non è necessario attivare questa funzionalità se:

- I tuoi utenti accedono solo con operazioni API autenticate come AdminInitiateAuth , e tu usi il `ContextData` parametro.
- Vuoi che le operazioni API non autenticate inviino un'impronta del dispositivo, ma non un indirizzo IP, alle funzionalità di sicurezza avanzata di Amazon Cognito.

Aggiorna il client dell'app come indicato di seguito nella console Amazon Cognito per aggiungere il supporto per ulteriori dati utente contestuali.

1. Accedi alla console Amazon Cognito.
2. Nel riquadro di navigazione, scegli Manage your User Pools (Gestisci i tuoi bacini d'utenza) e scegli i bacini d'utenza che intendi modificare.
3. Scegli la scheda App integration (Integrazione app).
4. In App clients and analytics (Client di app e analisi dei dati), scegli o crea un client dell'app. Per ulteriori informazioni, consulta Configurazione di un client dell'app per un pool di utenti.
5. Scegli Edit (Modifica) nel container App client information (Informazioni sul client dell'app).
6. Nell'area Advanced authentication settings (Impostazioni di autenticazione avanzate) per il client dell'app, scegli Accept additional user context data (Accetta dati utente contestuali aggiuntivi).
7. Seleziona Salvataggio delle modifiche.

Per configurare il client dell'app in modo che accetti i dati contestuali dell'utente nell'API Amazon Cognito, imposta su `EnablePropagateAdditionalUserContextData` `true` in una richiesta CreateUserPoolClientor UpdateUserPoolClient. Per informazioni sull'attivazione della sicurezza avanzata dall'applicazione Web o dai dispositivi mobili, consulta Attivazione della sicurezza avanzata del pool di utenti dall'app. Quando l'app chiama Amazon Cognito dal server, raccogli i dati di contesto degli utenti dal lato client. Di seguito è riportato un esempio che utilizza il metodo JavaScript SDK. `getData`

```
var encodedData =
 AmazonCognitoAdvancedSecurityData.getData(username, userPoolId, clientId);
```

Quando progetti la tua app per utilizzare l'autenticazione adattiva, ti consigliamo di integrare l'ultimo SDK Amazon Cognito nell'app. L'ultima versione dell'SDK raccoglie informazioni di impronta digitale del dispositivo come ID, modello e fuso orario del dispositivo. Per ulteriori informazioni sugli SDK di Amazon Cognito, consulta Installazione di un SDK del bacino d'utenza. La funzionalità di sicurezza avanzata di Amazon Cognito salva e assegna un punteggio di rischio solo agli eventi inviati dall'app nel formato corretto. Se Amazon Cognito restituisce una risposta di errore, controlla che la richiesta includa un hash segreto valido e che il parametro `IPaddress` sia un indirizzo IPv4 o IPv6 valido.

Risorse **`ContextData`** e **`UserContextData`**

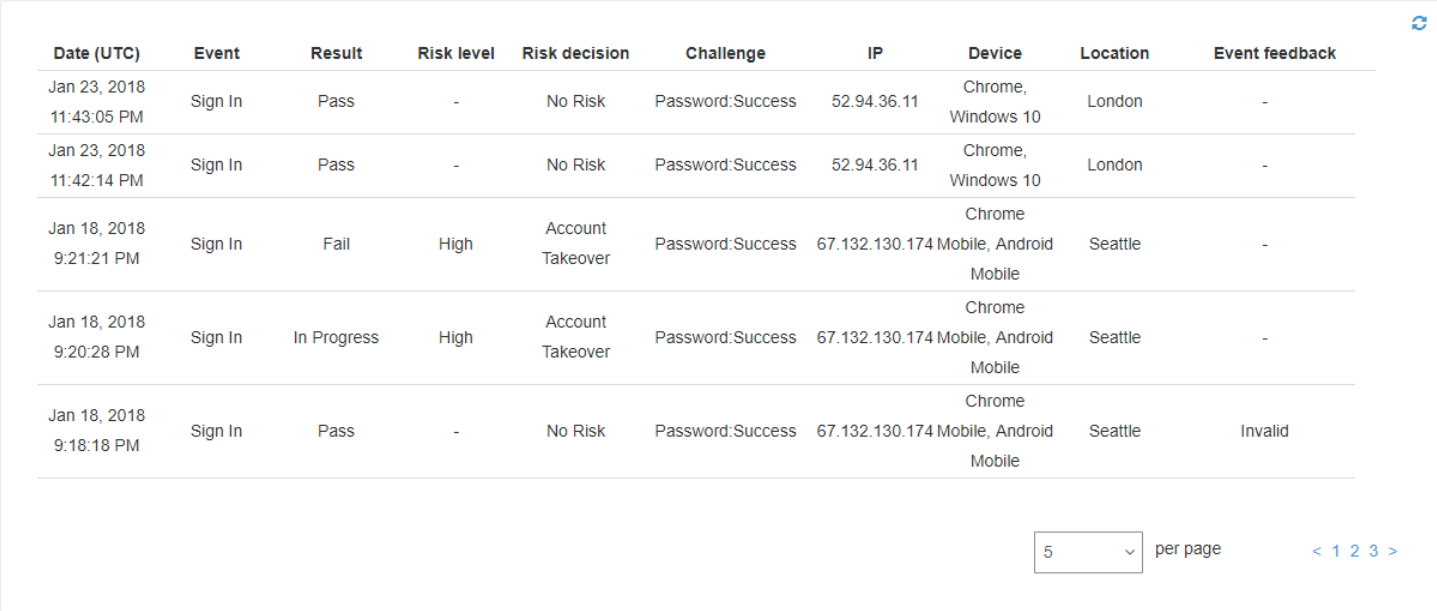
- AWS Amplify SDK per Android: GetUserContextData

- AWS Amplify SDK per iOS: userContextData
- JavaScript: amazon-cognito-advanced-security -data.min.js

Visualizzazione della cronologia eventi dell'utente

> **Note**
>
> Nella nuova console di Amazon Cognito, puoi visualizzare la cronologia degli eventi dell'utente nella scheda Users (Utenti).

Per consultare la cronologia degli accessi di un utente, puoi scegliere l'utente nella scheda Users (Utenti) della console Amazon Cognito. Amazon Cognito mantiene la cronologia eventi dell'utente per due anni.



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 >

Ogni evento di accesso ha un ID evento. L'evento contiene anche dati di contesto corrispondenti, come la posizione, i dettagli del dispositivo e i risultati del rilevamento dei rischi. Puoi interrogare la cronologia degli eventi degli utenti con l'operazione dell'API Amazon Cognito AdminListUserAuthEventso con AWS Command Line Interface (AWS CLI) with admin-list-user-auth -events.

Puoi anche correlare l'ID evento con il token emesso da Amazon Cognito nel momento in cui ha registrato l'evento. Gli ID e i token di accesso includono questo ID evento nel payload. Amazon Cognito mette inoltre in correlazione l'utilizzo del token di aggiornamento con l'ID evento originale.

Puoi tenere traccia dell'ID evento originale risalendo fino all'ID dell'evento di accesso che ha avuto come esito il rilascio dei token di Amazon Cognito. Puoi tracciare l'utilizzo di un token all'interno del sistema di autenticazione fino a un determinato evento di autenticazione. Per ulteriori informazioni, consulta Utilizzo di token con bacini d'utenza.

Fornitura del feedback sugli eventi

Il feedback sugli eventi influisce in tempo reale sulla valutazione dei rischi e migliora l'algoritmo di valutazione dei rischi nel corso del tempo. Puoi fornire un feedback sulla validità dei tentativi di accesso tramite la console Amazon Cognito e le operazioni delle API.

> **Note**
>
> Il tuo feedback sugli eventi influisce sul livello di rischio che Amazon Cognito assegna alle sessioni utente successive con le stesse caratteristiche.

Nella console Amazon Cognito, scegli un utente dalla scheda Users (Utenti) e seleziona Provide event feedback (Fornisci un feedback sugli eventi). Puoi controllare i dettagli dell'evento e quindi scegliere Set as valid (Imposta come valido) o Set as invalid (Imposta come non valido).

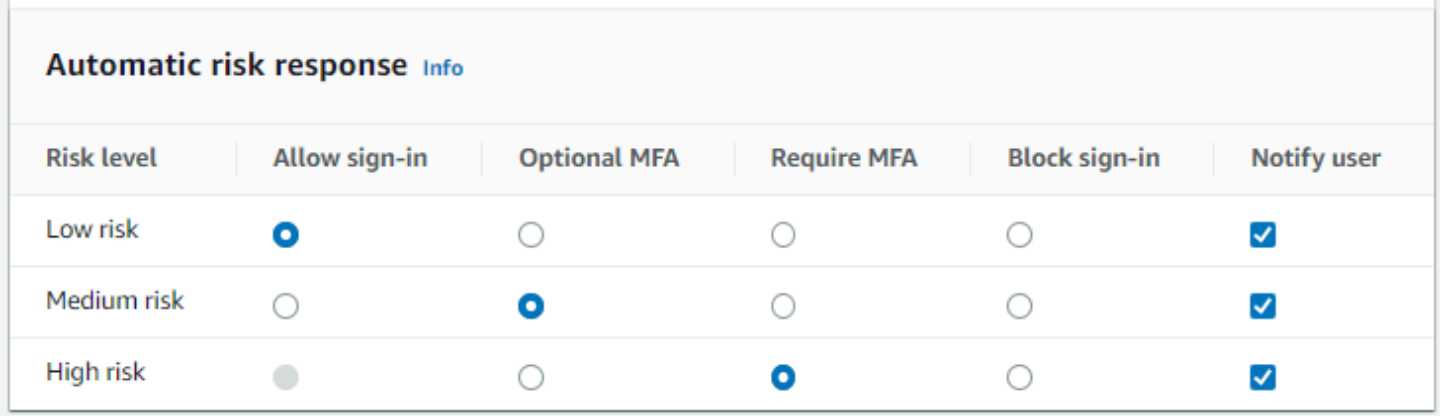
La console elenca la cronologia degli accessi nella scheda Users and groups (Utenti e gruppi). Se selezioni una voce, puoi contrassegnare l'evento come valido o non valido. Puoi anche fornire feedback tramite l'operazione API del pool di utenti e AdminUpdateAuthEventFeedbackil AWS CLI comando admin-update-auth-event -feedback.

Quando nella console Amazon Cognito selezioni Set as valid (Imposta come valido) o nell'API fornisci il valore `valid` per `FeedbackValue`, indichi ad Amazon Cognito che la sessione utente che ha valutato con un certo livello di rischio è invece attendibile. Quando nella console Amazon Cognito selezioni Set as invalid (Imposta come non valido) o nell'API fornisci il valore `invalid` per `FeedbackValue`, indichi ad Amazon Cognito che la sessione utente non è attendibile o che non ha valutato un livello di rischio sufficientemente elevato.

Invio di messaggi di notifica

Grazie alla protezione della sicurezza avanzata, Amazon Cognito può inviare notifiche agli utenti riguardo ai tentativi di accesso rischioso. Amazon Cognito può anche chiedere agli utenti di selezionare un link per indicare se l'accesso è valido o non valido. Amazon Cognito utilizza questo feedback per migliorare l'accuratezza del rilevamento del rischio per il tuo bacino d'utenza

Nella sezione Automatic risk response (Risposta automatica al rischio) scegli Notify Users (Notifica agli utenti) per i casi di basso, medio e alto rischio.



Amazon Cognito invia le notifiche e-mail agli utenti a prescindere dalla verifica dell'indirizzo e-mail.

Puoi personalizzare i messaggi e-mail di notifica e fornire i messaggi nelle versioni in formato di testo normale e in formato HTML. Per personalizzare le notifiche e-mail, apri nella configurazione della sicurezza avanzata Email templates (Modelli di e-mail) da Adaptive authentication messages (Messaggi di autenticazione adattiva). Per ulteriori informazioni sui modelli di e-mail, consulta Modelli dei messaggi.

Visualizzazione dei parametri di sicurezza avanzata

Amazon Cognito pubblica metriche per funzionalità di sicurezza avanzate sul tuo account in Amazon. CloudWatch Amazon Cognito raggruppa i parametri di sicurezza avanzata per livello di rischio e anche per livello di richiesta.

Per visualizzare le metriche nella console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri).
3. Scegli Amazon Cognito.
4. Scegli un gruppo di parametri aggregati, ad esempio Classificazione per rischio.
5. Nella scheda All metrics (Tutti i parametri) sono visualizzati tutti i parametri per quella scelta. Puoi eseguire le operazioni indicate di seguito:

   - Per ordinare la tabella, utilizza l'intestazione della colonna.

- Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.
- Per filtrare in base a una risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).
- Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

Parametro	Descrizione	Dimensioni parametro
CompromisedCredentialRisk	Richieste in cui Amazon Cognito ha rilevato credenziali compromesse.	Operazione: il tipo di operazione `PasswordChange` , `SignIn` o `SignUp` sono le uniche dimensioni. UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto (impostazione predefinita), medio o basso.
AccountTakeoverRisk	Richieste in cui Amazon Cognito ha rilevato il rischio di furto dell'account.	Operazione: il tipo di operazione `PasswordChange` , `SignIn` o `SignUp` sono le uniche dimensioni. UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto, medio o basso.
OverrideBlock	Richieste che Amazon Cognito ha bloccato a causa della configurazione fornita dallo sviluppatore.	Operazione: il tipo di operazione `PasswordChange` , `SignIn` o `SignUp` sono le uniche dimensioni.

Parametro	Descrizione	Dimensioni parametro
		UserPoolId: l'identificatore del pool di utenti. RiskLevel: alto, medio o basso.
Rischio	Richieste contrassegnate da Amazon Cognito come rischiose.	Operazione: tipo di operazione come `PasswordChange`, `SignIn` o `SignUp`. UserPoolId: l'identificatore del pool di utenti.
NoRisk	Richieste in cui Amazon Cognito non ha identificato alcun rischio.	Operazione: tipo di operazione come `PasswordChange`, `SignIn` o `SignUp`. UserPoolId: l'identificatore del pool di utenti.

Amazon Cognito ti offre due gruppi predefiniti di metriche per un'analisi pronta. CloudWatch Per classificazione in base al rischio si intende la granularità del livello di rischio per le richieste che Amazon Cognito identifica come rischiose. La classificazione in base alla richiesta riflette le metriche aggregate per livello di richiesta.

Gruppo di parametri aggregati	Descrizione
Classificazione per rischio	Richieste identificate da Amazon Cognito come rischiose.
Classificazione per richiesta	Parametri aggregati per richiesta.

Attivazione della sicurezza avanzata del bacino d'utenza dall'app

Dopo aver configurato le caratteristiche di sicurezza avanzata per il bacino d'utenza, devi abilitarle nell'app Web o mobile.

Utilizzo della sicurezza avanzata con JavaScript

1. Aggiungi l'SDK Amazon Cognito Identity per JavaScript alla tua app.
2. In CognitoUserPool.js, imposta su. `AdvancedSecurityDataCollectionFlag` `true` Imposta `UserPoolId` sull'ID del pool di utenti.
3. Aggiungi questo riferimento alla fonte al JavaScript file della tua app. Regione AWS Sostituiscilo `<region>` con uno dei seguenti:`us-east-1`,`us-east-2`,`us-west-2`,`eu-west-1`,`eu-west-2`, o`eu-central-1`.

```
<script src="https://amazon-cognito-assets.<region>.amazoncognito.com/amazon-
cognito-advanced-security-data.min.js"></script>
```

Utilizzo della sicurezza avanzata con Android

1. Crea la tua app con AWS Amplify per Android. Per ulteriori informazioni, consultare Configurazione del progetto nel AWS Amplify Dev Center.
2. Con `userContextDataProvider`, includi le informazioni su utenti e dispositivi nelle richieste di autenticazione.

Per informazioni sull'aggiunta di dati del contesto utente nell'SDK Android precedente, consultare aws-android-sdk-cognitoidentityprovider-asf.

Utilizzo della sicurezza avanzata con iOS

1. Crea la tua app con AWS Amplify for Swift o Flutter. Per ulteriori informazioni, consultare Swift Project Setup e Flutter Project Setup nel AWS Amplify Dev Center.
2. Includi le informazioni su utenti e dispositivi nelle richieste di autenticazione. Per un esempio da utilizzare con l'operazione InitiateAuthAPI, vedi `userContextData` in InitiateAuthInput +Amplify.swift on. GitHub

Per informazioni sull'aggiunta di dati del contesto utente nell'SDK iOS precedente, consultare AWSCognitoIdentityProviderASF.

Associazione di un ACL Web a un pool di utenti AWS WAF

AWS WAF è un firewall per applicazioni Web. Con una AWS WAF lista di controllo degli accessi Web (Web ACL), puoi proteggere il tuo pool di utenti da richieste indesiderate all'interfaccia utente ospitata

e agli endpoint del servizio API Amazon Cognito. Una ACL Web offre un controllo granulare su tutte le richieste Web HTTPS a cui risponde il pool di utenti. Per ulteriori informazioni sugli ACL AWS WAF Web, consulta Gestione e utilizzo di una lista di controllo degli accessi Web (Web ACL) nella Guida per gli sviluppatori.AWS WAF

Quando hai un ACL AWS WAF Web associato a un pool di utenti, Amazon Cognito inoltra le intestazioni e i contenuti selezionati non riservati delle richieste degli utenti a. AWS WAF AWS WAF ispeziona il contenuto della richiesta, la confronta con le regole che hai specificato nell'ACL web e restituisce una risposta ad Amazon Cognito.

Cose da sapere sugli ACL AWS WAF Web e Amazon Cognito

- Le richieste bloccate da AWS WAF non vengono conteggiate ai fini della quota di richiesta per nessun tipo di richiesta. Il AWS WAF gestore viene chiamato prima dei gestori di limitazione a livello di API.
- Quando si crea una ACL Web, passa del tempo prima che l'ACL Web si propaghi completamente e sia disponibile per Amazon Cognito. Il tempo di propagazione può variare da pochi secondi a diversi minuti. AWS WAF restituisce a `WAFUnavailableEntityException`quando si tenta di associare un ACL Web prima che si sia completamente propagato.
- È possibile associare una ACL Web a un pool di utenti.
- La richiesta potrebbe comportare un payload superiore ai limiti di quanto AWS WAF può controllare. Consulta Gestione dei componenti di richieste sovradimensionate nella AWS WAF Developer Guide per scoprire come configurare come AWS WAF gestire le richieste sovradimensionate da Amazon Cognito.
- Non puoi associare un ACL Web che utilizza la prevenzione dell'acquisizione di account AWS WAF Fraud Control (ATP) a un pool di utenti Amazon Cognito. La caratteristica ATP viene implementata quando si aggiunge il gruppo di regole gestito `AWS-AWSManagedRulesATPRuleSet`. Prima di associarla a un pool di utenti, assicurarsi che l'ACL Web non utilizzi questo gruppo di regole gestito.
- Quando hai un ACL AWS WAF web associato a un pool di utenti e una regola nell'ACL web presenta un CAPTCHA, ciò può causare un errore irreversibile nella registrazione TOTP dell'interfaccia utente ospitata. Per creare una regola contenente un'azione CAPTCHA, ma che non abbia alcun effetto sul token di software TOTP dell'interfaccia utente ospitata, consulta Configurazione dell'ACL AWS WAF Web per l'interfaccia utente TOTP MFA ospitata.

AWS WAF esamina le richieste ai seguenti endpoint.

Interfaccia utente ospitata

Richieste a tutti gli endpoint in Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata.

Operazioni API pubbliche

Richieste dalla tua app all'API Amazon Cognito che non utilizzano AWS credenziali per l'autorizzazione. Ciò include operazioni API come InitiateAuth, e RespondToAuthChallenge. GetUser Le operazioni API incluse nell'ambito di AWS WAF non richiedono l'autenticazione con AWS credenziali. Non sono autenticate o sono autorizzate con una stringa di sessione o un token di accesso. Per ulteriori informazioni, consulta Operazioni API autenticate e non autenticate per pool di utenti di Amazon Cognito.

È possibile configurare le regole nell'ACL Web con operazioni di regole Count (Conteggio), Allow (Consenti), Block (Blocca), oppure che presentano un CAPTCHA in risposta a una richiesta che corrisponde a una regola. Per ulteriori informazioni, consulta Regole personalizzate AWS WAF nella Guida per gli sviluppatori di AWS WAF . A seconda dell'operazione della regola, è possibile personalizzare la risposta che Amazon Cognito restituisce agli utenti.

Important

Le opzioni per personalizzare la risposta all'errore dipendono dal modo in cui si effettua una richiesta API.

- È possibile personalizzare il codice di errore e il corpo della risposta delle richieste dell'interfaccia utente ospitata. È possibile presentare un CAPTCHA che l'utente possa risolvere solo nell'interfaccia utente ospitata.
- Per le richieste effettuate con l'API dei pool di utenti di Amazon Cognito, puoi personalizzare il corpo della risposta di una richiesta che riceve una risposta Blocca. È inoltre possibile specificare un codice di errore personalizzato compreso tra 400 e 499.
- AWS Command Line Interface (AWS CLI) e gli AWS SDK restituiscono un `ForbiddenException` errore alle richieste che producono una risposta Block o CAPTCHA.

Associazione di una ACL Web al pool di utenti

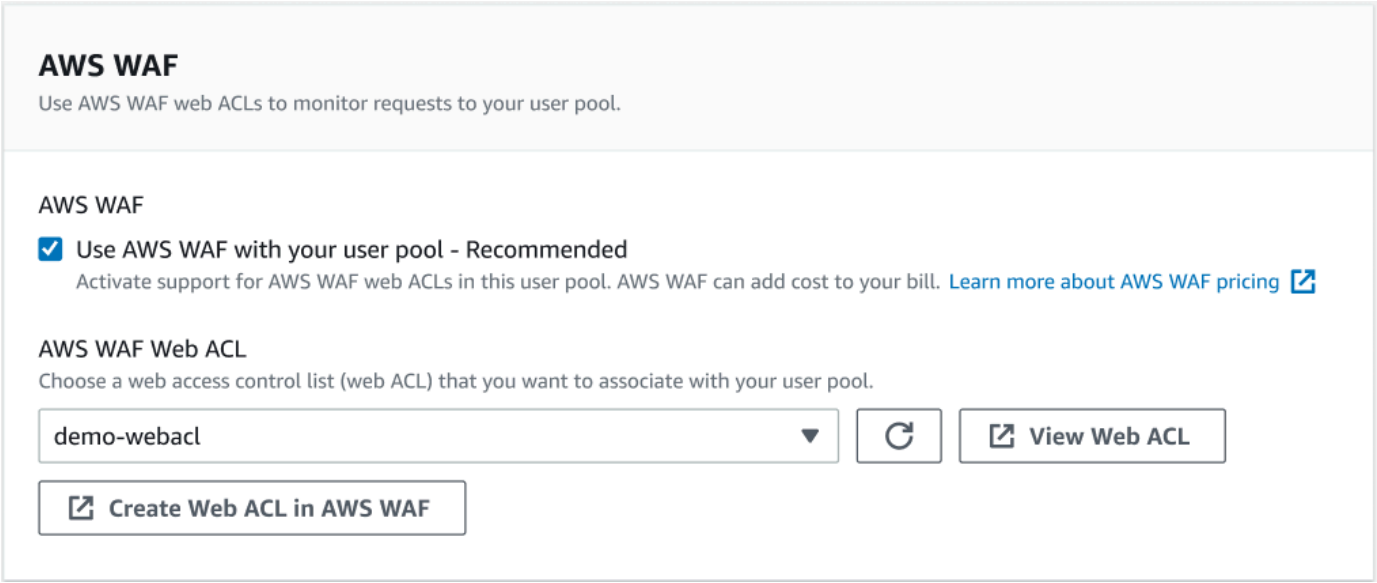
Per utilizzare un ACL Web nel tuo pool di utenti, il tuo principale AWS Identity and Access Management (IAM) deve disporre delle seguenti autorizzazioni Amazon Cognito. Per informazioni sulle AWS WAF autorizzazioni, consulta le autorizzazioni AWS WAF API nella Guida per gli sviluppatori.AWS WAF

- `cognito-idp:AssociateWebACL`
- `cognito-idp:DisassociateWebACL`
- `cognito-idp:GetWebACLForResource`
- `cognito-idp:ListResourcesForWebACL`

Sebbene sia necessario concedere le autorizzazioni IAM, le azioni elencate fanno riferimento solo alle autorizzazioni e non corrispondono a un'operazione API.

Per attivarlo AWS WAF per il tuo pool di utenti e associare un ACL web

1. Accedi alla console Amazon Cognito.
2. Nel pannello di navigazione, scegli User Pools (Bacini d'utenza) e seleziona i bacini d'utenza che intendi modificare.
3. Scegliere la scheda User pool properties (Proprietà del pool di utenti).
4. Scegliere Edit (Modifica) accanto a AWS WAF.
5. In AWS WAF, seleziona Usa AWS WAF con il tuo pool di utenti.



6. Scegliete un ACL AWS WAF Web già creato oppure scegliete Crea ACL Web in AWS WAF per crearne uno in una nuova AWS WAF sessione in. AWS Management Console
7. Seleziona Salvataggio delle modifiche.

Per associare a livello di codice un ACL Web al tuo pool di utenti in AWS Command Line Interface o in un SDK, utilizza ACL dell'API. AssociateWeb AWS WAF Amazon Cognito non dispone di un'operazione API distinta che associ una lista ACL Web.

Test e registrazione degli ACL web AWS WAF

Quando imposti un'azione della regola su Count nell'ACL web, AWS WAF aggiunge la richiesta a un conteggio di richieste che corrispondono alla regola. Per testare una ACL Web con il pool di utenti, impostare le operazioni delle regole su Count (Conteggio) e considerare il volume di richieste che corrispondono a ciascuna regola. Ad esempio, se una regola che desideri impostare sull'operazione Block (Blocca) corrisponde a un gran numero di richieste che si ritiene sia un normale traffico utente, potrebbe essere necessario riconfigurare la regola. Per ulteriori informazioni, consulta Testare e ottimizzare AWS WAF le protezioni nella Guida per gli AWS WAF sviluppatori.

Puoi anche configurare AWS WAF per registrare le intestazioni delle richieste in un gruppo di log di Amazon CloudWatch Logs, un bucket Amazon Simple Storage Service (Amazon S3) o un Amazon Data Firehose. Puoi identificare le richieste Amazon Cognito effettuate con l'API dei pool di utenti tramite `x-amzn-cognito-client-id` e `x-amzn-cognito-operation-name`. Le richieste di interfaccia utente ospitate includono solo l'intestazione `x-amzn-cognito-client-id`. Per ulteriori informazioni, consulta Registrazione del traffico ACL Web nella Guida per gli sviluppatori di AWS WAF .

AWS WAF gli ACL web non sono soggetti ai prezzi delle funzionalità di sicurezza avanzate di Amazon Cognito. Le funzionalità di sicurezza AWS WAF completano le funzionalità di sicurezza avanzate di Amazon Cognito. Puoi attivare entrambe le funzionalità in un pool di utenti. AWS WAF fattura separatamente per l'ispezione delle richieste del pool di utenti. Per ulteriori informazioni, consulta la sezione Prezzi di AWS WAF.

I dati della AWS WAF richiesta di registrazione sono soggetti a una fatturazione aggiuntiva da parte del servizio a cui si indirizzano i log. Per ulteriori informazioni, consulta Prezzi per la registrazione delle informazioni sul traffico ACL web nella Guida per gli sviluppatori di AWS WAF .

Distinzione tra maiuscole e minuscole del bacino d'utenza

Per impostazione predefinita, i pool di utenti di Amazon Cognito creati in non AWS Management Console fanno distinzione tra maiuscole e minuscole. Quando un bacino d'utenza è senza distinzione tra maiuscole e minuscole, user@example.com e User@example.com si riferiscono allo stesso utente. Quando i nomi utente in un pool di utenti sono senza distinzione tra maiuscole e minuscole, anche gli attributi `preferred_username` e `email` sono senza distinzione tra maiuscole e minuscole.

Per tenere conto delle impostazioni di sensibilità alle maiuscole e alle minuscole del bacino d'utenza, identifica gli utenti nel codice dell'app in base a un attributo utente alternativo. Dal momento che il caso di un attributo nome utente, nome utente preferito o indirizzo e-mail può variare a seconda dei profili utente, fai riferimento all'attributo `sub`. È inoltre possibile creare un attributo personalizzato immutabile nel bacino d'utenza e assegnare il proprio valore identificativo univoco all'attributo in ogni nuovo profilo utente. Quando crei per la prima volta un utente, puoi scrivere un valore in un attributo personalizzato immutabile.

> **Note**
>
> Indipendentemente dalle impostazioni di distinzione tra maiuscole e minuscole del bacino d'utenza, Amazon Cognito richiede che un utente federato di un provider di identità SAML oppure OIDC passi un'attestazione `NameId` o `sub` univoca e con distinzione tra maiuscole e minuscole. Per ulteriori informazioni sull'identificatore univoco, sulla distinzione tra maiuscole e minuscole e IdPs SAML, consulta. Utilizzo dell'accesso SAML avviato da SP

Creazione di un bacino d'utenza con distinzione tra maiuscole e minuscole

Se create risorse con le operazioni AWS Command Line Interface (AWS CLI) e API, ad esempio CreateUserPool, dovete impostare il parametro `CaseSensitive` Boolean su. `false` Questa impostazione crea un bacino d'utenza senza distinzione tra maiuscole e minuscole. Se non specifichi un valore, il `CaseSensitive` viene impostato di default su `true`. Questo comportamento è opposto a quello per i bacini d'utenza creati nel AWS Management Console. Prima del 12 febbraio 2020, i bacini d'utenza sono stati impostati di default con distinzione tra maiuscole e minuscole, a prescindere dalla piattaforma.

Puoi utilizzare la scheda Esperienza di accesso AWS Management Console o l'operazione DescribeUserPoolAPI per esaminare le impostazioni relative alla distinzione tra maiuscole e minuscole per ogni pool di utenti del tuo account.

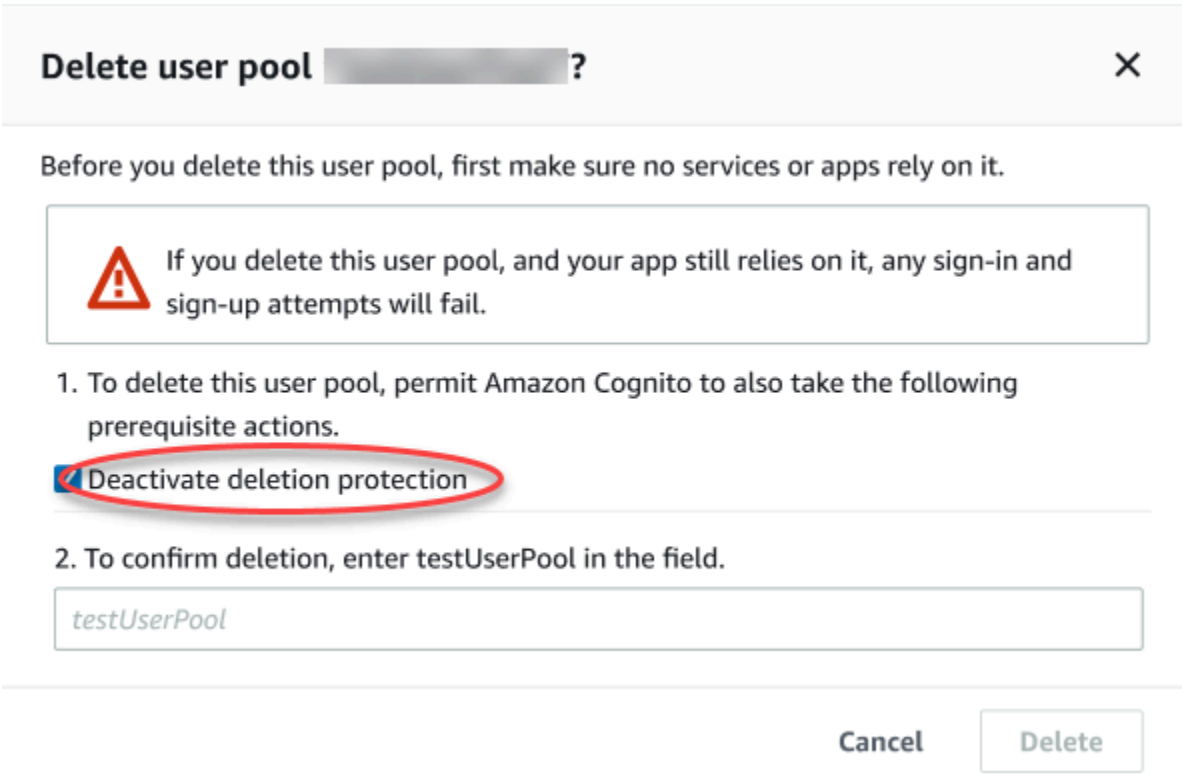
Migrazione a un nuovo bacino d'utenza

A causa di potenziali conflitti tra i profili utente, non puoi modificare un bacino d'utenza Amazon Cognito da sensibille alle maiuscole e alle minuscole a insensibile alle maiuscole e alle minuscole. Invece, migra gli utenti in un nuovo bacino d'utenza. È necessario creare un codice di migrazione per risolvere i conflitti correlati ai casi. Questo codice deve restituire un nuovo utente univoco o rifiutare il tentativo di accesso quando rileva un conflitto. In un nuovo bacino d'utenza senza distinzione tra maiuscole e minuscole, assegna un Trigger Lambda di migrazione utenti. La AWS Lambda funzione può creare utenti nel nuovo pool di utenti senza distinzione tra maiuscole e minuscole. Quando l'utente non riesce ad accedere con il bacino d'utenza senza distinzione tra maiuscole e minuscole, la funzione Lambda trova e duplica l'utente dal bacino d'utenza con distinzione tra maiuscole e minuscole. Puoi anche attivare un trigger ForgotPasswordLambda per la migrazione degli utenti sugli eventi. Amazon Cognito trasmette le informazioni sull'utente e i metadati degli eventi dall'azione di accesso o ripristino della password alla funzione Lambda. Puoi utilizzare i dati dell'evento per gestire conflitti tra nomi utente e indirizzi e-mail quando la funzione crea il nuovo utente nel pool di utenti senza distinzione tra maiuscole e minuscole. Questi conflitti si verificano tra nomi utente e indirizzi e-mail che sarebbero univoci in un pool di utenti senza distinzione tra maiuscole e minuscole, ma sono identici in un pool di utenti con distinzione tra maiuscole e minuscole.

Per ulteriori informazioni su come utilizzare un trigger Lambda di utenti di migrazione tra pool di utenti Amazon Cognito, consulta Migrazione degli utenti ai pool di utenti di Amazon Cognito nel blog. AWS

Protezione da eliminazione del bacino d'utenza

Per evitare che gli amministratori eliminino accidentalmente il pool di utenti, attiva la protezione dall'eliminazione. Con la protezione dalle eliminazioni attiva, è necessario confermare che si desidera eliminare il pool di utenti prima di eliminarlo. Quando si elimina un pool di utenti in AWS Management Console, è possibile disattivare contemporaneamente la protezione dall'eliminazione. Quando accetti la richiesta di disattivare la protezione dall'eliminazione e confermi l'intenzione di eliminare, come mostrato nell'immagine seguente, Amazon Cognito elimina il tuo pool di utenti.



Quando desideri eliminare un pool di utenti con una richiesta API Amazon Cognito, devi prima cambiare `DeletionProtection` in `Inactive` a una richiesta UpdateUserPool. Se non disattivi la protezione dall'eliminazione, Amazon Cognito restituisce un errore `InvalidParameterException`. Dopo aver disattivato la protezione dall'eliminazione, è possibile eliminare il pool di utenti in una richiesta DeleteUserPool.

Amazon Cognito attiva Deletion protection (Protezione dall'eliminazione) per impostazione predefinita quando crei un nuovo pool di utenti nella AWS Management Console. Quando crei un pool di utenti con l'API `CreateUserPool`, la protezione dall'eliminazione è inattiva per impostazione predefinita. Per utilizzare questa funzionalità nei pool di utenti creati con la AWS CLI o un SDK AWS, imposta il parametro `DeletionProtection` su `True`.

Puoi attivare o disattivare lo stato di protezione dall'eliminazione nel container di Deletion protection (Protezione dall'eliminazione) nella scheda User pool settings (Impostazioni del pool di utenti) della console Amazon Cognito.

Per configurare la protezione da eliminazione

1. Passa alla console Amazon Cognito. Potrebbe comparire una richiesta di inserimento delle credenziali AWS.

2. Scegli User Pools (bacini d'utenza).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda Impostazioni del pool di utenti. Individua Protezione da eliminazione e seleziona Attiva o Disattiva.
5. Conferma la tua scelta nella finestra di dialogo successiva.

Gestione delle risposte agli errori relativi all'esistenza degli utenti

Amazon Cognito supporta la personalizzazione delle risposte di errore restituite dai pool di utenti. Sono disponibili risposte di errore personalizzate per le operazioni di creazione e autenticazione utente, ripristino password e conferma.

Utilizza l'impostazione `PreventUserExistenceErrors` di un client di applicazioni di un bacino d'utenza per abilitare o disabilitare errori relativi all'esistenza degli utenti. Quando crei una nuova app, il client con l'API dei pool di utenti di Amazon Cognito `PreventUserExistenceErrors` è `LEGACY` o disabilitato per impostazione predefinita. Nella console Amazon Cognito, l'opzione Previeni gli errori di esistenza degli utenti, un'impostazione di `ENABLED` for, `PreventUserExistenceErrors` è selezionata per impostazione predefinita. Per aggiornare la `PreventUserExistenceErrors` configurazione, esegui una delle seguenti operazioni:

- Modifica il valore `PreventUserExistenceErrors` tra `ENABLED` e `LEGACY` in una richiesta UpdateUserPoolClientAPI.
- Modifica il client dell'app nella console Amazon Cognito e modifica lo stato di Previeni gli errori di esistenza degli utenti tra selezionato (`ENABLED`) e deselezionato (). `LEGACY`

Se questa proprietà ha un valore di`LEGACY`, il client dell'app restituisce una risposta di `UserNotFoundException` errore quando un utente tenta di accedere con un nome utente che non esiste nel tuo pool di utenti.

Quando questa proprietà ha un valore di`ENABLED`, il client dell'app non rivela l'inesistenza di un account utente nel pool di utenti con un `UserNotFoundException` errore. Una `PreventUserExistenceErrors` configurazione di `ENABLED` ha i seguenti effetti:

- Amazon Cognito risponde con informazioni non specifiche alle richieste API laddove la sua risposta potrebbe altrimenti rivelare l'esistenza di un utente valido.

- Le API di accesso e password dimenticata di Amazon Cognito restituiscono una risposta generica di errore di autenticazione. La risposta di errore indica che il nome utente o la password non sono corretti.
- Le API di conferma dell'account e ripristino della password di Amazon Cognito restituiscono una risposta che indica che un codice è stato inviato a un supporto di consegna simulato, anziché una rappresentazione parziale delle informazioni di contatto di un utente.

Le seguenti informazioni descrivono in dettaglio i comportamenti delle operazioni del pool di utenti quando `PreventUserExistenceErrors` è impostato su. `ENABLED`

Operazioni di autenticazione e creazione degli utenti

È possibile configurare le risposte di errore sia nell'autenticazione Username-password che nell'autenticazione Secure Remote Password (SRP). È inoltre possibile personalizzare gli errori restituiti con l'autenticazione personalizzata. Le seguenti API eseguono queste operazioni di autenticazione:

- `AdminInitiateAuth`
- `AdminRespondToAuthChallenge`
- `InitiateAuth`
- `RespondToAuthChallenge`

L'elenco seguente mostra come personalizzare le risposte agli errori nelle operazioni di autenticazione utente.

Autenticazione con nome utente e password

Per consentire l'accesso di un utente con `ADMIN_USER_PASSWORD_AUTH` e `USER_PASSWORD_AUTH`, includi il nome utente e la password in una richiesta API `AdminInitiateAuth` o `InitiateAuth`. Amazon Cognito restituisce un errore `NotAuthorizedException` generico quando il nome utente o la password non sono corretti.

Autenticazione basata su password remota protetta (SRP)

Per consentire l'accesso di un utente con `USER_SRP_AUTH`, includi un nome utente e un parametro `SRP_A` in una richiesta API `AdminInitiateAuth` o `InitiateAuth`. In risposta, Amazon Cognito restituisce `SRP_B` e aggiorna l'utente. Quando un utente non viene trovato, Amazon Cognito restituisce una risposta simulata nella prima fase, come descritto in RFC 5054.

Amazon Cognito restituisce lo stesso "salt" e un ID utente interno in formato UUID (Universally Unique Identifier) per la stessa combinazione di nome utente e pool di utenti. Quando si invia una richiesta API `RespondToAuthChallenge` con verifica della password, Amazon Cognito restituisce un errore `NotAuthorizedException` generico quando il nome utente o la password non sono corretti.

> **Note**
>
> È possibile simulare una risposta generica con autenticazione con nome utente e password se si utilizzano attributi alias basati sulla verifica e il formato del nome utente immutabile non è un UUID.

Trigger Lambda di richieste di autenticazione personalizzate

Se utilizzi Trigger Lambda di richieste di autenticazione personalizzate e abiliti le risposte di errore, allora `LambdaChallenge` restituisce un parametro booleano denominato `UserNotFound`. Quindi viene inviato nella richiesta dei trigger Lambda `DefineAuthChallenge`, `VerifyAuthChallenge` e `CreateAuthChallenge`. Puoi utilizzare questo trigger per simulare le richieste di autenticazione personalizzate per un utente che non esiste. Se richiami il trigger Lambda pre-autenticazione per un utente che non esiste, Amazon Cognito restituisce `UserNotFound`.

L'elenco seguente mostra come personalizzare le risposte di errore nelle operazioni di creazione utente.

SignUp

L'`SignUp`operazione ritorna sempre `UsernameExistsException` quando è già in uso un nome utente. Per evitare che Amazon Cognito restituisca un errore `UsernameExistsException` per indirizzi e-mail e numeri di telefono durante la registrazione di utenti nell'app, utilizza gli attributi alias basati sulla verifica. Per ulteriori informazioni sugli alias, consultare Personalizzazione degli attributi di accesso.

Per un esempio di come Amazon Cognito può impedire l'uso delle richieste API `SignUp` per scoprire gli utenti del pool di utenti in uso, consulta Prevenzione degli errori `UsernameExistsException` relativi a indirizzi e-mail e numeri di telefono durante la registrazione.

Utenti importati

Se `PreventUserExistenceErrors` è abilitato, durante l'autenticazione degli utenti importati viene restituito un errore `NotAuthorizedException` generico che indica che il nome utente o la password non erano corretti, anziché restituire `PasswordResetRequiredException`. Consulta Necessità degli utenti importati di ripristinare le loro password per maggiori informazioni.

Trigger Lambda di migrazione utenti

Amazon Cognito restituirà una risposta simulata per gli utenti inesistenti quando il trigger Lambda imposta una risposta vuota nel contesto dell'evento originale. Per ulteriori informazioni, consulta Trigger Lambda di migrazione utenti.

Prevenzione degli errori **`UsernameExistsException`** relativi a indirizzi e-mail e numeri di telefono durante la registrazione

Nell'esempio seguente viene illustrato come impedire che indirizzi e-mail e numeri di telefono duplicati generino errori `UsernameExistsException` in risposta alle richieste API `SignUp`, durante la configurazione di attributi alias nel pool di utenti. È necessario aver creato il pool di utenti con indirizzo e-mail o numero di telefono come un attributo alias. Per ulteriori informazioni, consulta la sezione Personalizzazione degli attributi di accesso di Attributi del pool di utenti.

1. Jie esegue la registrazione per un nuovo nome utente e fornisce anche l'indirizzo e-mail `jie@example.com`. Amazon Cognito invia un codice all'indirizzo e-mail.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp sign-up --client-id 1234567890abcdef0 --username jie --password
    PASSWORD --user-attributes Name="email",Value="jie@example.com"
   ```

   Example response

   ```
   {
       "UserConfirmed": false,
       "UserSub": "<subId>",
       "CodeDeliveryDetails": {
           "AttributeName": "email",
           "Destination": "j****@e****",
           "DeliveryMedium": "EMAIL"
       }
   ```

```
}
```

2. Jie fornisce il codice inviato per confermare di essere il titolare dell'indirizzo email. Questo completa la registrazione come un utente.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=jie --
   confirmation-code xxxxxx
   ```

3. Shirley registra un nuovo account utente e fornisce l'indirizzo e-mail `jie@example.com`. Amazon Cognito non restituisce un errore `UsernameExistsException` e invia un codice di conferma all'indirizzo e-mail di Jie.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp sign-up --client-id 1234567890abcdef0 --username shirley --password
    PASSWORD --user-attributes Name="email",Value="jie@example.com"
   ```

   Example response

   ```
   {
       "UserConfirmed": false,
       "UserSub": "<new subId>",
       "CodeDeliveryDetails": {
           "AttributeName": "email",
           "Destination": "j****@e****",
           "DeliveryMedium": "EMAIL"
       }
   }
   ```

4. In uno scenario diverso, Shirley è la titolare di `jie@example.com`. Shirley recupera il codice inviato da Amazon Cognito all'indirizzo e-mail di Jie e tenta di confermare l'account.

   AWS CLI Comando di esempio

   ```
   aws cognito-idp confirm-sign-up --client-id 1234567890abcdef0 --username=shirley --
   confirmation-code xxxxxx
   ```

   Example response

```
An error occurred (AliasExistsException) when calling the ConfirmSignUp operation: An
 account with the email already exists.
```

Amazon Cognito non restituisce un errore alla richiesta `aws cognito-idp sign-up` di Shirley, nonostante `jie@example.com` sia stato assegnato a un utente esistente. Shirley deve dimostrare di essere titolare dell'indirizzo e-mail prima che Amazon Cognito restituisca una risposta di errore. In un pool di utenti con attributi alias, questo comportamento impedisce l'uso dell'API `SignUp` pubblica per verificare se esiste un utente con un determinato indirizzo e-mail o numero di telefono.

Questo comportamento è diverso dalla risposta restituita da Amazon Cognito alla richiesta `SignUp` con un nome utente esistente, come illustrato nell'esempio seguente. Sebbene Shirley apprenda da questa risposta che esiste già un utente con il nome utente `jie`, non viene informata di alcun indirizzo e-mail o numero di telefono associato all'utente.

Comando della CLI di esempio

```
aws cognito-idp sign-up --client-id 1example23456789 --username jie --password PASSWORD
      --user-attributes Name="email",Value="shirley@example.com"
```

Example response

```
An error occurred (UsernameExistsException) when calling the SignUp operation: User
 already exists
```

Operazioni di ripristino della password

Amazon Cognito restituisce le seguenti risposte alle operazioni di reimpostazione della password utente quando si evitano errori relativi all’esistenza dell'utente.

ForgotPassword

> Se un utente non viene trovato, è disattivato o non dispone di un meccanismo di consegna verificato per recuperare la password, Amazon Cognito restituisce `CodeDeliveryDetails` con un supporto di recapito simulato per un utente. Il supporto di recapito simulato è determinato dal formato del nome utente di input e dalle impostazioni di verifica del pool di utenti.

ConfirmForgotPassword

Amazon Cognito restituisce l'errore `CodeMismatchException` per gli utenti che non esistono o sono disabilitati. Se non viene richiesto un codice quando si utilizza `ForgotPassword`, Amazon Cognito restituisce l'errore `ExpiredCodeException`.

Operazioni di conferma

Amazon Cognito restituisce le seguenti risposte alle operazioni di conferma e verifica quando si evitano errori relativi all’esistenza dell'utente.

ResendConfirmationCode

Amazon Cognito restituisce l'errore `CodeDeliveryDetails` per un utente disabilitato o uno che non esiste. Per un utente esistente, Amazon Cognito invia un codice di conferma all'indirizzo e-mail o al telefono dell'utente.

ConfirmSignUp

`ExpiredCodeException` restituisce se un codice è scaduto. Amazon Cognito restituisce `NotAuthorizedException` quando un utente non è autorizzato. Se il codice non corrisponde a quello previsto dal server, Amazon Cognito restituisce `CodeMismatchException`.

Pool di identità di Amazon Cognito

Un pool di identità di Amazon Cognito è una directory di identità federate che puoi scambiare per credenziali AWS . I pool di identità generano AWS credenziali temporanee per gli utenti della tua app, indipendentemente dal fatto che abbiano effettuato l'accesso o che tu non li abbia ancora identificati. Con i ruoli e le policy AWS Identity and Access Management (IAM), puoi scegliere il livello di autorizzazione che desideri concedere ai tuoi utenti. Gli utenti possono iniziare come ospiti e recuperare le risorse mantenute nei Servizi AWS. Quindi, possono accedere con un provider di identità di terze parti per sbloccare l'accesso alle risorse che vengono rese disponibili ai membri registrati. Il provider di identità di terze parti può essere un provider OAuth 2.0 utente (social) come Apple o Google, un provider di identità SAML o OIDC personalizzato o uno schema di autenticazione personalizzato, chiamato anche provider degli sviluppatori, di propria progettazione.

Funzionalità del pool di identità di Amazon Cognito

Firma le richieste per Servizi AWS

Firma le richieste API su Amazon Simple Storage Service (Amazon S3) e Amazon DynamoDB. Servizi AWS Analizza l'attività degli utenti con servizi come Amazon Pinpoint e Amazon. CloudWatch

Filtrare richieste con policy basate su risorse

Esercita controllo granulare sull'accesso utente alle risorse. Trasforma le richieste dell'utente in tag della sessione IAM e crea policy IAM che concedono l'accesso alle risorse a sottoinsiemi distinti di utenti.

Assegnare l'accesso guest

Per gli utenti che non hanno ancora effettuato l'accesso, configura il pool di identità per generare credenziali AWS con un ambito di accesso ristretto. Autentica gli utenti tramite un provider Single Sign-On per migliorarne l'accesso.

Assegnare ruoli IAM in base alle caratteristiche degli utenti

Assegna un singolo ruolo IAM a tutti gli utenti autenticati o scegli il ruolo in base alle richieste di ciascun utente.

Accettare un'ampia gamma di provider di identità

Scambia un ID o un token di accesso, un token del pool di utenti, un'asserzione SAML o un token OAuth di un social provider con credenziali. AWS

Convalidare le proprie identità

Esegui la convalida degli utenti e utilizza le tue credenziali di sviluppatore AWS per emettere credenziali per i tuoi utenti.

Potrebbe essere già disponibile un pool di utenti Amazon Cognito che fornisce servizi di autenticazione e autorizzazione all'app. Puoi configurare il pool di utenti come gestore dell'identità digitale per il pool di identità. Quando lo fai, i tuoi utenti possono autenticarsi tramite il tuo pool di utenti IdPs, consolidare le loro affermazioni in un token di identità OIDC comune e scambiare quel token con credenziali. AWS L'utente può quindi presentare le proprie credenziali in una richiesta firmata ai Servizi AWS.

Puoi anche presentare richieste autenticate da uno qualsiasi dei provider di identità direttamente al pool di identità. Amazon Cognito personalizza le dichiarazioni degli utenti dei provider SAML, OAuth e OIDC in una richiesta API per credenziali a breve termine. AssumeRoleWithWebIdentity

I pool di utenti Amazon Cognito sono analoghi ai provider di identità OIDC per le app abilitate per SSO. I pool di identità fungono da provider di identità AWS per qualsiasi app con dipendenze di risorse che funzionano meglio con l'autorizzazione IAM.

I pool di identità di Amazon Cognito supportano i seguenti provider di identità:

- Provider pubblici: Configurazione di Login with Amazon come IdP di pool di identità, Configurazione di Facebook come pool di identità IdP, Configurazione di Google come IdP del pool di identità, Configurazione di Accedi con Apple come IdP del pool di identità, Twitter.
- Pool di utenti Amazon Cognito
- Configurazione di un provider OIDC come pool di identità IdP
- Configurazione di un provider SAML come IdP del pool di identità
- Identità autenticate dagli sviluppatori (pool di identità)

Per informazioni sulla disponibilità regionale dei pool di identità di Amazon Cognito, consulta Disponibilità delle regioni del servizio AWS.

Per ulteriori informazioni sui pool di identità di Amazon Cognito, consulta gli argomenti riportati di seguito.

Argomenti

- Utilizzo dei pool di identità (identità federate)
- Concetti del pool di identità
- Best practice di sicurezza per i pool di identità di Amazon Cognito
- Utilizzo di attributi per il controllo degli accessi
- Utilizzo del controllo degli accessi basato su ruoli
- Ottenere le credenziali
- Accesso ai servizi AWS
- Provider di identità esterni con pool di identità
- Identità autenticate dagli sviluppatori (pool di identità)
- Cambio degli utenti non autenticati in utenti autenticati (pool di identità)

Utilizzo dei pool di identità (identità federate)

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per gli utenti ospiti (non autenticati) e per gli utenti che sono stati autenticati e hanno ricevuto un token. Un pool di identità è un archivio di dati sull'identità dell'utente specifici per il tuo account.

Per creare un nuovo pool di identità nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità.
2. Scegli Crea pool di identità.
3. In Configurazione dell'attendibilità del pool di identità, scegli di configurare il pool di identità per Accesso autenticato, Accesso guest o entrambi.
    - Se hai scelto Accesso autenticato, seleziona uno o più Tipi di identità che desideri impostare come origine delle identità autenticate nel pool di identità. Se configuri un Provider degli sviluppatori personalizzato, non puoi modificarlo né eliminarlo dopo aver creato il pool di identità.
4. In Configura le autorizzazioni, scegli un ruolo IAM predefinito per gli utenti autenticati o guest nel pool di identità.
    a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza il documento di

policy per esaminare le autorizzazioni che verranno assegnate da Amazon Cognito al nuovo ruolo IAM.

b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

5. In Connect identity providers, inserisci i dettagli dei provider di identità (IdPs) che hai scelto in Configure identity pool trust. È possibile che ti venga chiesto di fornire informazioni sul client dell'app OAuth, scegliere un pool di utenti Amazon Cognito, scegliere un IdP IAM o inserire un identificatore personalizzato per un provider degli sviluppatori.

   a. Scegli le impostazioni del ruolo per ogni IdP. Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.

      i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

      ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

   b. Configura Attributi per il controllo degli accessi per ciascun IdP. L'opzione Attributi per il controllo degli accessi associa le richieste dell'utente ai tag principali applicati da Amazon Cognito alla relativa sessione temporanea. Puoi creare policy IAM per filtrare l'accesso utente in base ai tag applicati alla relativa sessione.

      i. Per non applicare alcun tag principale, scegli Inattivo.

      ii. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

iii. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

6. In Configura proprietà, inserisci un Nome in Nome del pool di identità.
7. In Autenticazione di base (classica), scegli Attiva flusso di base, se desiderato. Con il flusso di base attivo, puoi ignorare le selezioni di ruolo che hai effettuato per te IdPs e chiamare AssumeRoleWithWebIdentitydirettamente. Per ulteriori informazioni, consulta Flusso di autenticazione dei pool di identità (identità federate).
8. In Tag, scegli Aggiungi tag se desideri applicare tag al pool di identità.
9. In Esamina e crea, conferma le selezioni effettuate per il nuovo pool di identità. Seleziona Modifica per tornare alla procedura guidata e modificare le eventuali impostazioni. Al termine, seleziona Crea un pool di identità.

Ruoli IAM dell'utente

Un ruolo IAM definisce le autorizzazioni per gli utenti di accedere a AWS risorse, come. Amazon Cognito Sync Gli utenti dell'applicazione assumeranno i ruoli da te creati. Puoi specificare diversi ruoli per utenti autenticati e non autenticati. Per ulteriori informazioni sui ruoli IAM, consulta Ruoli IAM.

Identità autenticate e non autenticate

I pool di identità di Amazon Cognito supportano sia le identità autenticate che le identità non autenticate. Le identità autenticate appartengono agli utenti autenticati da qualsiasi provider di identità supportato. Solitamente le identità non autenticate appartengono a utenti guest.

- Per configurare le identità autenticate con un provider di accesso pubblico, consulta Provider di identità esterni con pool di identità.
- Per configurare il tuo processo di autenticazione di back-end, consulta Identità autenticate dagli sviluppatori (pool di identità).

Attivazione o disattivazione dell'accesso guest

L'accesso degli ospiti ai pool di identità di Amazon Cognito (identità non autenticate) fornisce un identificatore e AWS credenziali univoci per gli utenti che non si autenticano con un provider di identità. Se l'applicazione consente utenti che non effettuano l'accesso, puoi attivare l'accesso per

le identità non autenticate. Per ulteriori informazioni, consulta Guida introduttiva ai pool di identità di Amazon Cognito.

Per aggiornare l'accesso ospite in un pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Accesso guest. In un pool di identità che attualmente non supporta l'accesso guest, Stato è Inattivo.

    a. Se Accesso guest è Attivo e desideri disattivarlo, seleziona Disattiva.

    b. Se Accesso guest è Inattivo e desideri attivarlo, seleziona Modifica.

    - Scegli un ruolo IAM predefinito per gli utenti guest nel pool di identità.

        A. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza il documento di policy per esaminare le autorizzazioni che verranno assegnate da Amazon Cognito al nuovo ruolo IAM.

        B. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo che desideri utilizzare. Account AWS Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.

        C. Seleziona Salva modifiche.

        D. Per attivare l'accesso guest, seleziona Attiva nella scheda Accesso utente.

Modifica del ruolo associato a un tipo di identità

Ogni identità nel pool di identità può essere autenticata o non autenticata. Le identità autenticate appartengono agli utenti che sono autenticati da un provider di accesso pubblico (bacini d'utenza di Amazon Cognito, Login with Amazon, Accedi con Apple, Facebook, Google, SAML o qualsiasi

provider OpenID Connect) o un provider per gli sviluppatori (il tuo processo di autenticazione back-end). Solitamente le identità non autenticate appartengono a utenti guest.

Per ogni tipo di identità, esiste un ruolo assegnato. A questo ruolo è associata una politica che stabilisce a quale Servizi AWS ruolo può accedere. Quando Amazon Cognito riceve una richiesta, il servizio determina il tipo di identità e il ruolo assegnato a quel tipo di identità e utilizza la policy associata a tale ruolo per rispondere. Modificando una politica o assegnando un ruolo diverso a un tipo di identità, puoi controllare a quale tipo Servizi AWS di identità può accedere. Per visualizzare o modificare le policy associate ai ruoli nel pool di identità, vedi la Console AWS IAM.

Per modificare il ruolo predefinito autenticato o non autenticato del pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Accesso guest o Accesso autenticato. In un pool di identità che non è attualmente configurato per tale tipo di accesso, Stato è Inattivo. Seleziona Edit (Modifica).
4. Scegli un ruolo IAM predefinito per gli utenti guest o autenticati nel pool di identità.

    a. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza il documento di policy per esaminare le autorizzazioni che verranno assegnate da Amazon Cognito al nuovo ruolo IAM.

    b. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.
5. Seleziona Salva modifiche.

Modifica dei provider di identità

Se consenti agli utenti di effettuare l'autenticazione utilizzando i provider di identità utente (ad esempio pool di utenti Amazon Cognito, Login with Amazon, Accedi con Apple, Facebook o Google), puoi specificare gli identificatori dell'applicazione nella console dei pool di identità Amazon Cognito

(identità federate). Questo associa l'ID dell'applicazione (fornito dal provider di accesso pubblico) al pool di identità.

Puoi anche configurare regole di autenticazione per ogni provider da questa pagina. Ogni provider consente fino a 25 regole. Le regole vengono applicate nell'ordine in cui le hai salvate per ogni provider. Per ulteriori informazioni, consulta Utilizzo del controllo degli accessi basato su ruoli.

> **Warning**
>
> La modifica dell'ID dell'applicazione IdP collegato nel pool di identità impedisce agli utenti esistenti di effettuare l'autenticazione con il pool di identità. Per ulteriori informazioni, consulta Provider di identità esterni con pool di identità.

Per aggiornare un gestore dell'identità digitale del pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare. Se desideri aggiungere un nuovo IdP, seleziona Aggiungi provider di identità.
   - Se hai scelto Aggiungi provider di identità, scegli uno dei Tipi di identità che desideri aggiungere.
4. Per modificare l'ID dell'applicazione, scegli Modifica in Informazioni sul provider di identità.
5. Per modificare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modificain Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure puoi selezionare l'opzione Scegli ruolo con regole. Con un IdP del pool di utenti Amazon Cognito, puoi anche scegliere un ruolo con preferred_role nei token. Per ulteriori informazioni sulla richiesta `cognito:preferred_role`, consultare Assegnazione dei valori di priorità ai gruppi.
     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

6. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modifica in Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

7. Seleziona Salva modifiche.

Eliminazione di un pool di identità

L'eliminazione del pool di identità non può essere annullata. Dopo aver eliminato un pool di identità, tutte le app e gli utenti che dipendono da esso smettono di funzionare.

Eliminazione di un pool di identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona il pulsante di opzione accanto al pool di identità che desideri eliminare.
2. Seleziona Elimina.
3. Inserisci o incolla il nome del pool di identità e seleziona Elimina.

> **Warning**
>
> Selezionando il pulsante di eliminazione, eliminerai il pool di identità e tutti i dati utente che contiene. L'eliminazione di un pool di identità interromperà il funzionamento delle applicazioni e degli altri servizi che utilizzano il pool di identità.

Eliminazione di un'identità da un pool di identità

Quando si elimina un'identità da un pool di identità, vengono rimosse le informazioni di identificazione archiviate da Amazon Cognito per tale utente federato. Quando l'utente richiede nuovamente le credenziali, riceve un nuovo ID identità se il pool di identità considera ancora attendibile il provider di identità. Questa operazione non può essere annullata.

Per eliminare un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona le caselle di controllo accanto alle identità che desideri eliminare e scegli Elimina. Conferma che desideri eliminare le identità e scegli Elimina.

Utilizzo di Amazon Cognito Sync con pool di identità

Amazon Cognito Sync è una Servizio AWS libreria client che consente di sincronizzare i dati utente relativi alle applicazioni tra dispositivi. Con Amazon Cognito Sync puoi sincronizzare i dati del profilo utente tra dispositivi mobili e il Web, senza utilizzare il tuo back-end. Le librerie client archiviano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati. Se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Gestione di set di dati

Se nell'applicazione hai implementato la funzionalità Amazon Cognito Sync, la console dei pool di identità di Amazon Cognito consente di creare ed eliminare manualmente set di dati e record per singole identità. Qualsiasi modifica apportata a set di dati o record di un'identità nella console dei pool di identità di Amazon Cognito non viene salvata finché non selezioniSynchronize (Sincronizza) nella console. La modifica non è visibile all'utente finale fino alla sincronizzazione delle chiamate dell'identità. I dati sincronizzati da altri dispositivi per le identità individuali sono visibili una volta aggiornata la pagina dei set di dati dell'elenco per un'identità in particolare.

Creazione di un set di dati per un'identità

Amazon Cognito Sync associa un set di dati a un'identità. Puoi compilare il set di dati con informazioni di identificazione sull'utente rappresentato dall'identità, quindi sincronizzare tali informazioni con tutti i dispositivi dell'utente.

Per aggiungere un set di dati e i record del set di dati a un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona l'identità da modificare.
4. In Set di dati, scegli Crea set di dati.
5. Inserisci un Nome set di dati e seleziona Crea set di dati.
6. Se desideri aggiungere record al set di dati, scegli il set di dati dai dettagli dell'identità. In Record, seleziona Crea record.
7. Inserisci una Chiave e un Valore per il record. Scegli Conferma. Ripeti l'operazione per aggiungere altri record.

Eliminazione di un set di dati associato a un'identità

Per eliminare un set di dati e i relativi record da un'identità

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Browser di identità.
3. Seleziona l'identità che contiene il set di dati da eliminare.
4. In Set di dati, scegli il pulsante di opzione accanto al set di dati da eliminare.
5. Seleziona Elimina. Esamina la scelta e seleziona nuovamente Elimina.

Pubblicazione in blocco di dati

La pubblicazione in blocco può essere utilizzata per esportare i dati già archiviati nell'archivio di Amazon Cognito Sync in un flusso Amazon Kinesis. Per istruzioni su come pubblicare in blocco tutti i flussi, consulta Amazon Cognito Streams.

Attivazione della sincronizzazione push

Amazon Cognito monitora automaticamente l'associazione tra identità e dispositivi. L'utilizzo della funzione di sincronizzazione push, assicura che ogni istanza di una determinata identità venga comunicata quando si modificano i dati di identità. La sincronizzazione push è tale per cui ogni volta che il set di dati cambia per un'identità, tutti i dispositivi associati a tale identità ricevono una notifica push silenziosa per segnalare le modifiche.

Puoi attivare la sincronizzazione push nella console di Amazon Cognito.

Per attivare la sincronizzazione push

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Scegli la scheda Proprietà del pool di identità.
3. In Sincronizzazione push, seleziona Modifica
4. Seleziona Attiva sincronizzazione push con il pool di identità.
5. Scegli una delle Applicazioni di piattaforma Amazon Simple Notification Service (Amazon SNS) che hai creato nella Regione AWS corrente. Amazon Cognito pubblica notifiche push nell'applicazione di piattaforma. Seleziona Crea applicazione di piattaforma per passare alla console di Amazon SNS e crearne una nuova.
6. Per pubblicare nell'applicazione di piattaforma, Amazon Cognito assume un ruolo IAM in Account AWS. Scegli Crea un nuovo ruolo IAM se desideri che Amazon Cognito crei automaticamente un nuovo ruolo con autorizzazioni di base e una relazione di affidabilità con il pool di identità. Inserisci un Nome ruolo IAM per identificare il nuovo ruolo, ad esempio `myidentitypool_authenticatedrole`. Seleziona Visualizza il documento di policy per esaminare le autorizzazioni che verranno assegnate da Amazon Cognito al nuovo ruolo IAM.
7. Puoi scegliere di utilizzare un ruolo IAM esistente se hai già un ruolo Account AWS che desideri utilizzare. Devi configurare la policy di attendibilità del ruolo IAM per includere `cognito-identity.amazonaws.com`. Configura la policy di attendibilità del ruolo per consentire ad Amazon Cognito di assumere il ruolo solo quando rende evidente che la richiesta ha avuto origine da un utente autenticato nel pool di identità specifico. Per ulteriori informazioni, consulta Attendibilità del ruolo e autorizzazioni.
8. Seleziona Salva modifiche.

Configurazione di Amazon Cognito Streams

Amazon Cognito Streams offre agli sviluppatori il controllo e l'analisi dei loro dati archiviati in Amazon Cognito Sync. Gli sviluppatori possono ora configurare un flusso Kinesis per ricevere eventi come dati. Amazon Cognito può eseguire il push di ogni modifica del set di dati in un flusso Kinesis in tempo reale. Per istruzioni su come configurare Amazon Cognito Streams nella console di Amazon Cognito, consulta Amazon Cognito Streams.

Configurazione di Amazon Cognito Events

Amazon Cognito Events ti consente di eseguire una AWS Lambda funzione in risposta a eventi importanti in Amazon Cognito Sync. Amazon Cognito Sync lancia l'evento trigger di sincronizzazione quando viene sincronizzato un set di dati. Puoi utilizzare l'evento del trigger di sincronizzazione per eseguire un'azione quando un utente aggiorna i dati. Per istruzioni sulla configurazione di Amazon Cognito Events dalla console, consulta Amazon Cognito Events.

Per ulteriori informazioni AWS Lambda, consulta. AWS Lambda

Concetti del pool di identità

I pool di identità di Amazon Cognito consentono di creare identità univoche per gli utenti e autenticarli con i provider di identità. Con un'identità, è possibile ottenere AWS credenziali temporanee con privilegi limitati per accedere ad altre. Servizi AWS I pool di identità di Amazon Cognito supportano i provider di identità pubbliche, come Amazon, Apple, Facebook e Google, nonché le identità non autenticate. Questa funzione supporta anche identità non autenticate per gli sviluppatori e ti consente di registrare e di autenticare gli utenti tramite il tuo processo di autenticazione di back-end.

Per informazioni sulla disponibilità regionale dei pool di identità di Amazon Cognito, consulta Disponibilità delle regioni del servizio AWS. Per ulteriori informazioni sui pool di identità di Amazon Cognito, consulta gli argomenti riportati di seguito.

Argomenti

- Flusso di autenticazione dei pool di identità (identità federate)
- Ruoli IAM
- Attendibilità del ruolo e autorizzazioni

Flusso di autenticazione dei pool di identità (identità federate)

Amazon Cognito consente di creare identificatori univoci per i tuoi utenti finali che sono mantenuti coerenti su tutti i dispositivi e le piattaforme. Amazon Cognito fornisce anche credenziali temporanee con privilegi limitati alla tua applicazione per accedere alle risorse. AWS In questa pagina sono riportati i concetti di base del funzionamento dell'autenticazione in Amazon Cognito e viene spiegato il ciclo di vita di un'identità all'interno del tuo pool di identità.

Flusso di autenticazione del provider esterno

Un utente che effettua l'autenticazione con Amazon Cognito passa attraverso un processo multifase per eseguire il bootstrap delle proprie credenziali. Amazon Cognito dispone di due flussi differenti per l'autenticazione con provider pubblici: avanzati e di base.

Una volta completato uno di questi flussi, puoi accedere ad altri, Servizi AWS come definito dalle politiche di accesso del tuo ruolo. Di default, la console Amazon Cognito crea ruoli con accesso all'archivio di Amazon Cognito Sync e ad Amazon Mobile Analytics. Per ulteriori informazioni su come concedere un accesso aggiuntivo, consulta Ruoli IAM.

I pool di identità accettano i seguenti elementi forniti dai provider:

Provider	Artefatto di autenticazione
Bacino d'utenza di Amazon Cognito	Token ID
OpenID Connect (OIDC)	Token ID
SAML 2.0	Asserzione SAML
Fornitore di servizi sociali	Token di accesso

Flusso di autenticazione avanzato (semplificato)

Quando utilizzi il flusso di autenticazione avanzato, l'app presenta innanzitutto una prova di autenticazione da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in GetIduna richiesta.

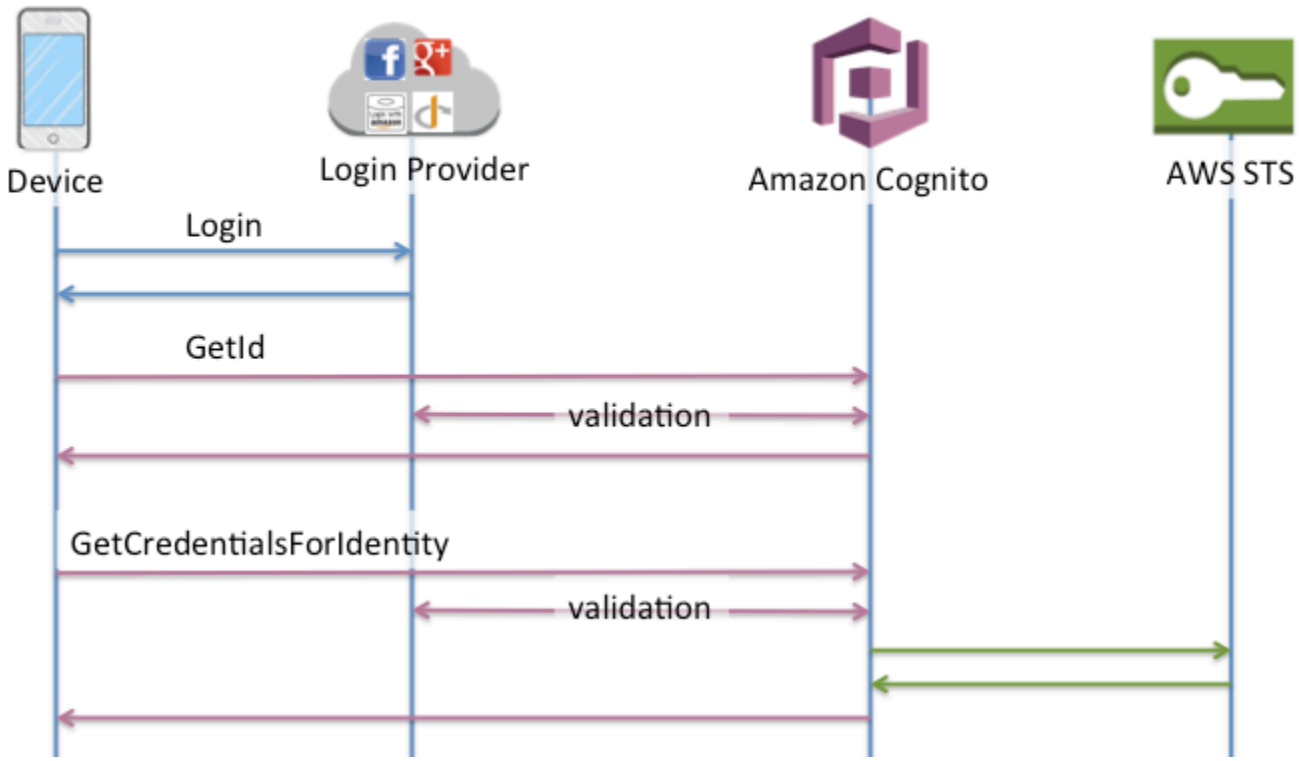
1. L'applicazione presenta una prova di autenticazione, un token web JSON o un'asserzione SAML, da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in una richiesta GetID.

2. Il tuo pool di identità restituisce un ID di identità.
3. L'applicazione combina l'ID di identità con la stessa prova di autenticazione in una GetCredentialsForIdentityrichiesta.
4. Il pool di identità restituisce AWS le credenziali.
5. La tua applicazione firma le richieste AWS API con le credenziali temporanee.

L'autenticazione avanzata gestisce la logica della selezione dei ruoli IAM e del recupero delle credenziali nella configurazione del pool di identità. Puoi configurare il tuo pool di identità per selezionare un ruolo predefinito, per applicare i principi del controllo degli accessi basato sugli attributi (ABAC) o del controllo degli accessi basato sui ruoli (RBAC) alla selezione dei ruoli. Le AWS credenziali dell'autenticazione avanzata sono valide per un'ora.

Ordine delle operazioni nell'autenticazione avanzata

1. `GetId`
2. `GetCredentialsForIdentity`



Flusso di autenticazione di base (classico)

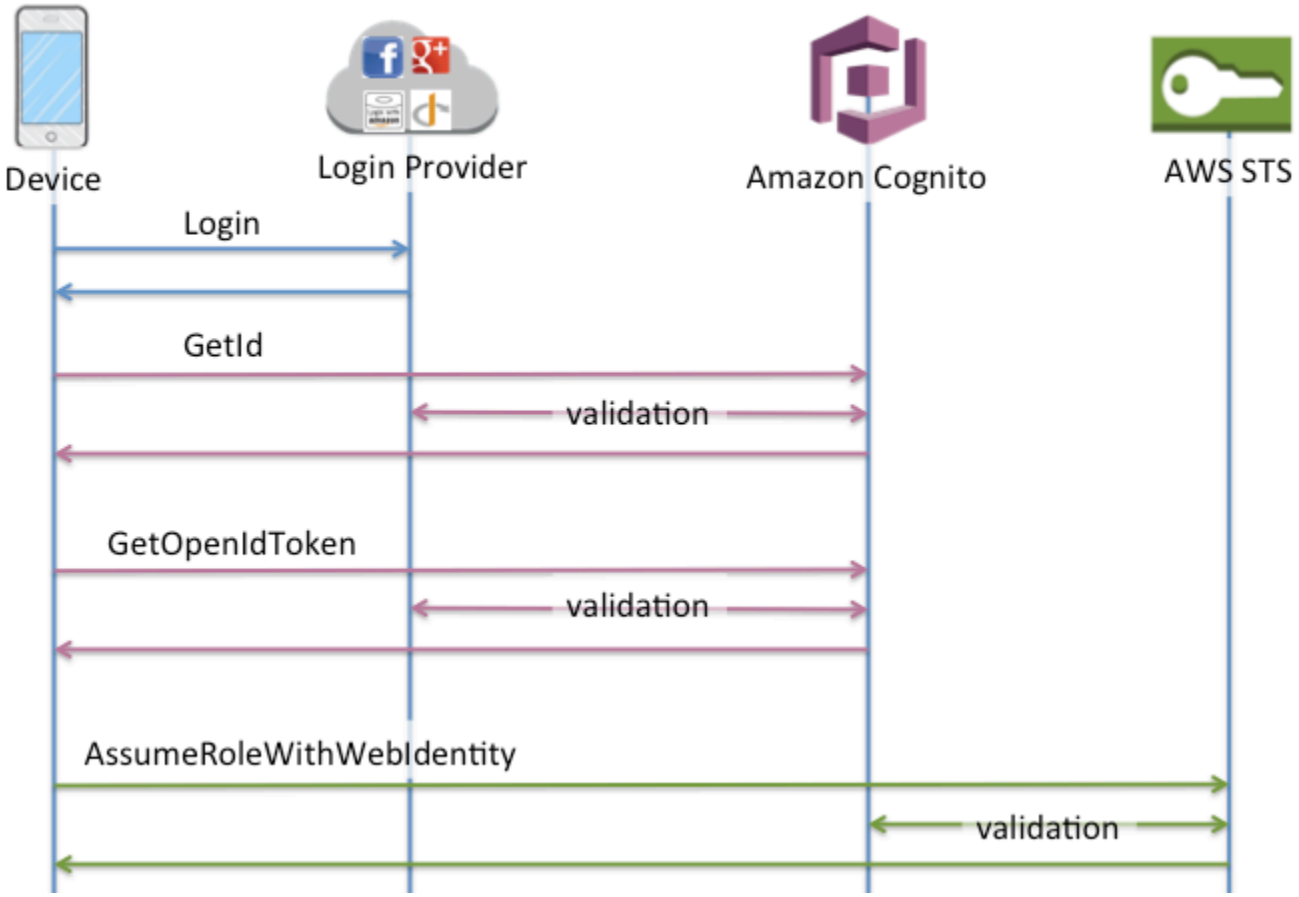
Quando si utilizza il flusso di autenticazione di base,

1. L'applicazione presenta una prova di autenticazione, un token web JSON o un'asserzione SAML, da un pool di utenti Amazon Cognito autorizzato o da un provider di identità di terze parti in una richiesta GetID.
2. Il tuo pool di identità restituisce un ID di identità.
3. L'applicazione combina l'ID di identità con la stessa prova di autenticazione in una GetOpenIdTokenrichiesta.
4. `GetOpenIdToken`restituisce un nuovo token OAuth 2.0 emesso dal tuo pool di identità.
5. L'applicazione presenta il nuovo token in una AssumeRoleWithWebIdentityrichiesta.
6. AWS Security Token Service AWS STS) restituisce AWS le credenziali.
7. L'applicazione firma le richieste AWS API con le credenziali temporanee.

Il flusso di lavoro di base offre un controllo più granulare sulle credenziali distribuite agli utenti. La richiesta `GetCredentialsForIdentity` del flusso di autenticazione avanzato richiede un ruolo basato sul contenuto di un token di accesso. La `AssumeRoleWithWebIdentity` richiesta nel flusso di lavoro classico garantisce all'app una maggiore capacità di richiedere le credenziali per qualsiasi AWS Identity and Access Management ruolo configurato con una politica di attendibilità sufficiente. Puoi anche richiedere una durata della sessione del ruolo personalizzata.

Ordine delle operazioni nell'autenticazione di base

1. `GetId`
2. `GetOpenIdToken`
3. `AssumeRoleWithWebIdentity`



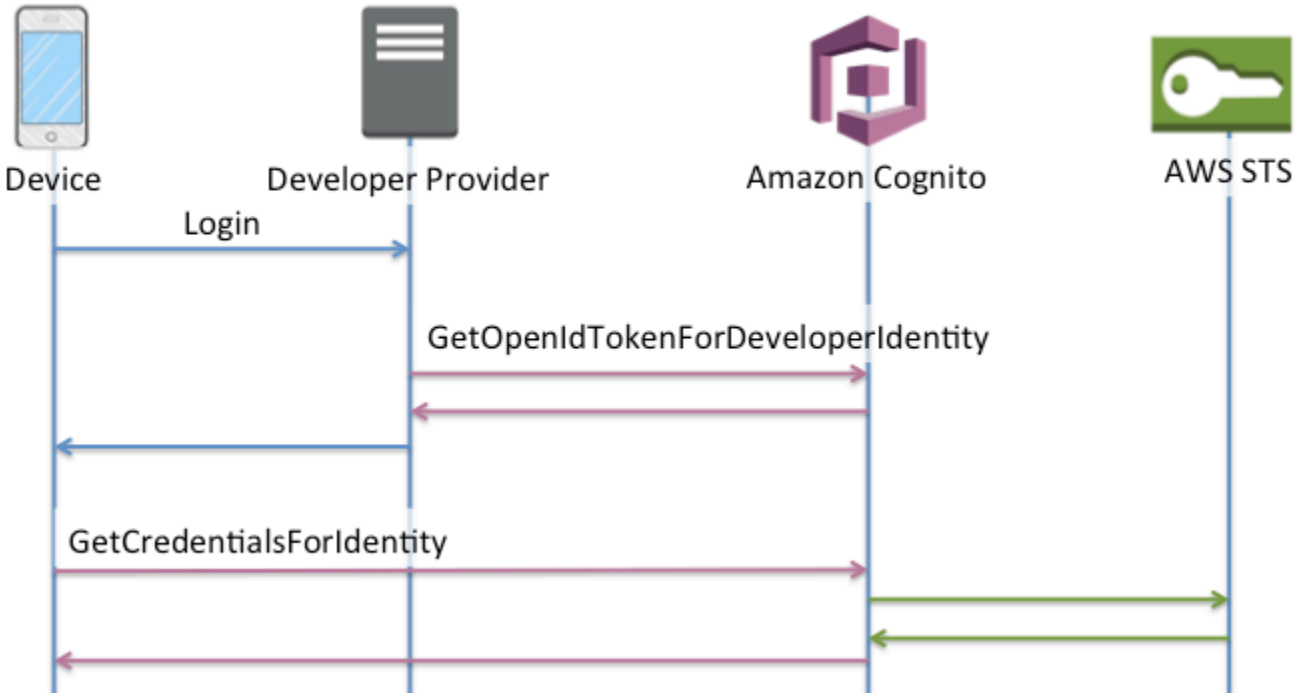
Flusso di autenticazione delle identità autenticate dagli sviluppatori

Quando utilizzi Identità autenticate dagli sviluppatori (pool di identità), il client usa un altro flusso di autenticazione che include il codice esterno ad Amazon Cognito per convalidare l'utente nel tuo sistema di autenticazione. Il codice al di fuori di Amazon Cognito è indicato come tale.

Flusso di autenticazione avanzato

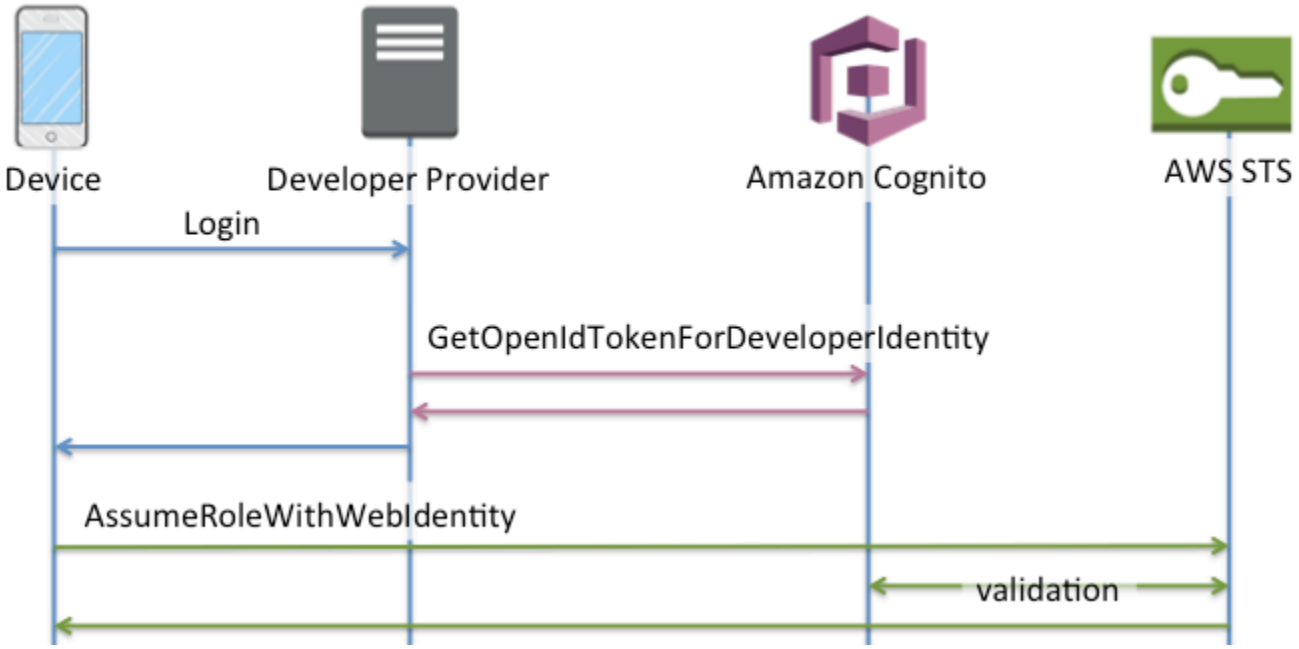
Ordine delle operazioni in Autenticazione avanzata con un provider di sviluppo

1. Accesso tramite provider per gli sviluppatori (codice esterno Amazon Cognito)
2. Convalida dell'accesso dell'utente (codice esterno ad Amazon Cognito)
3. GetOpenIdTokenForDeveloperIdentity
4. GetCredentialsForIdentity



Ordine delle operazioni nell'autenticazione di base con un provider di sviluppo

1. Implementa la logica all'esterno del pool di identità per accedere e generare un identificatore sviluppatore-provider.
2. Recupera le credenziali memorizzate sul lato server. AWS
3. Invia l'identificatore del provider di sviluppo in una richiesta GetOpenIdTokenForDeveloperIdentityAPI firmata con credenziali autorizzate. AWS
4. Richiedi le credenziali dell'applicazione con. AssumeRoleWithWebIdentity



Quale flusso di autenticazione usare?

Il flusso avanzato è la scelta più sicura con il più basso livello di impegno da parte degli sviluppatori:

- Il flusso migliorato riduce la complessità, le dimensioni e la frequenza delle richieste API.
- Non è necessario che l'applicazione effettui richieste API aggiuntive a AWS STS.
- Il tuo pool di identità valuta gli utenti in base alle credenziali del ruolo IAM che dovrebbero ricevere. Non è necessario incorporare la logica per la selezione dei ruoli nel cliente.

> **Important**
>
> Quando crei un nuovo pool di identità, non attivare l'autenticazione di base (classica) per impostazione predefinita, come best practice. Per implementare l'autenticazione di base, valuta innanzitutto le relazioni di fiducia dei tuoi ruoli IAM per le identità web. Quindi incorpora la logica per la selezione dei ruoli nel tuo client e proteggi il client dalle modifiche da parte degli utenti.

Il flusso di autenticazione di base delega la logica della selezione dei ruoli IAM alla tua applicazione. In questo flusso, Amazon Cognito convalida la sessione autenticata o non autenticata dell'utente ed emette un token con cui puoi scambiare credenziali. AWS STS Gli utenti possono scambiare i token

dell'autenticazione di base con qualsiasi ruolo IAM che si fida del tuo pool di identità e/o dello stato autenticato/non autenticato. `amr`

Allo stesso modo, tenete presente che l'autenticazione degli sviluppatori è una scorciatoia per la convalida dell'autenticazione del provider di identità. Amazon Cognito considera attendibili le AWS credenziali che autorizzano una GetOpenIdTokenForDeveloperIdentityrichiesta senza un'ulteriore convalida del contenuto della richiesta. Proteggi i segreti che autorizzano l'autenticazione degli sviluppatori dall'accesso da parte degli utenti.

Riepilogo delle API

GetId

La chiamata GetIdAPI è la prima chiamata necessaria per stabilire una nuova identità in Amazon Cognito.

Accesso non autenticato

Amazon Cognito può concedere le autorizzazione di accesso alle tue applicazioni agli utenti guest non autenticati. Se questa caratteristica è abilitata nel pool di identità, gli utenti possono richiedere un nuovo ID identità in qualsiasi momento tramite l'API `GetId`. È previsto che l'applicazione memorizzi nella cache questo ID identità per effettuare chiamate successive a Amazon Cognito. Gli SDK per AWS dispositivi mobili e l' AWS SDK for JavaScript in the Browser dispongono di fornitori di credenziali che gestiscono questa memorizzazione nella cache per te.

Accesso autenticato

Una volta che hai configurato la tua applicazione con il supporto per un provider di accesso pubblico (Facebook, Google +, Login with Amazon o Accedi con Apple), gli utenti sono anche in grado di fornire i token (OAuth o OpenID Connect) che li identificano in quei provider. Se utilizzato in una chiamata a `GetId`, Amazon Cognito crea una nuova identità autenticata oppure restituisce l'identità già associata con quel particolare accesso. Amazon Cognito fa ciò convalidando il token con il provider e assicurando che:

- Il token sia valido e che provenga dal provider configurato.
- Il token non sia scaduto.
- Il token corrisponda con l'identificatore dell'applicazione creato con quel provider (ad esempio,l'ID dell'app Facebook).
- Il token corrisponda con l'identificatore dell'utente.

GetCredentialsForIdentity

L'GetCredentialsForIdentityAPI può essere richiamata dopo aver stabilito un ID di identità. Questa operazione è funzionalmente equivalente alla chiamata GetOpenIdToken, quindi AssumeRoleWithWebIdentity.

Affinché Amazon Cognito possa chiamare `AssumeRoleWithWebIdentity` per tuo conto, il tuo bacino d'utenza deve avere i ruoli IAM ad esso associati. Puoi farlo tramite la console Amazon Cognito o manualmente tramite l'SetIdentityPoolRolesoperazione.

GetOpenIdToken

Effettua una richiesta GetOpenIdTokenAPI dopo aver stabilito un ID di identità. Memorizza l'ID di identità nella cache dopo la prima richiesta e avvia le successive sessioni di base (classiche) per l'identità con `GetOpenIdToken`.

La risposta a una richiesta API `GetOpenIdToken` è un token generato da Amazon Cognito. Puoi inviare questo token come `WebIdentityToken` parametro in una AssumeRoleWithWebIdentityrichiesta.

Prima di inviare il token OpenID, è necessario verificarlo nell'app. Puoi utilizzare le librerie OIDC dell'SDK o una libreria come aws-jwt-verify per verificare che Amazon Cognito abbia emesso il token. L'ID della chiave di firma, o `kid`, del token OpenID è uno di quelli elencati nel documento jwks_uri† dell'identità di Amazon Cognito. Queste chiavi sono soggette a modifiche. La funzione che verifica i token di identità di Amazon Cognito deve aggiornare periodicamente l'elenco delle chiavi dal documento jwks_uri. Amazon Cognito imposta la durata dell'aggiornamento nell'intestazione della risposta cache-control di jwks_uri, attualmente con `max-age` impostato su 30 giorni.

Accesso non autenticato

Per ottenere un token per un'identità non autenticata, necessiti soltanto dell'ID identità. Non è possibile ottenere un token non autenticato per identità autenticate o disattivate.

Accesso autenticato

Se disponi di un'identità autenticata, è necessario trasmettere almeno un token valido per un accesso già associato con quella identità. Tutti i token trasmessi durante la chiamata `GetOpenIdToken` dovranno trasmettere la stessa convalida menzionata in precedenza; se uno dei token fallisce, tutta la chiamata ha esito negativo. La risposta dalla chiamata `GetOpenIdToken` include anche l'ID identità. Questo avviene perché l'ID identità che trasmetti potrebbe non essere quello che viene restituito.

Collegamento degli accessi

Se invii un token per un accesso che non è già associato a un'identità, l'accesso è considerato "collegato" all'identità associata. Puoi collegare un solo accesso per provider pubblico. I tentavi di collegare più di un accesso con un provider pubblico determinano una risposta di errore `ResourceConflictException`. Se un accesso è semplicemente collegato a un'identità esistente, l'ID identità restituito dal `GetOpenIdToken` è uguale a quello che hai trasmesso.

Unione delle identità

Se trasmetti un token per un accesso che non è attualmente collegato a una determinata identità, bensì a un'altra, le due identità vengono unite. Una volta unite, un'identità diventa principale/proprietaria di tutti gli accessi associati e l'altra è disattivata. In questo caso, l'ID identità principale/proprietario viene restituito. È necessario aggiornare la cache locale se questo valore è diverso. I provider degli SDK AWS mobili o dell' AWS SDK per JavaScript il browser eseguono questa operazione per te.

GetOpenIdTokenForDeveloperIdentity

L'GetOpenIdTokenForDeveloperIdentityoperazione sostituisce l'uso di GetIde GetOpenIdTokendal dispositivo quando si utilizzano identità autenticate dagli sviluppatori. Poiché l'applicazione firma le richieste a questa operazione API con AWS credenziali, Amazon Cognito ritiene che l'identificatore utente fornito nella richiesta sia valido. L'autenticazione degli sviluppatori sostituisce la convalida dei token eseguita da Amazon Cognito con provider esterni.

Il payload per questa API include una mappa. `logins` Questa mappa deve contenere la chiave del provider di sviluppo e un valore come identificatore per l'utente nel sistema. Se l'identificatore dell'utente non è già collegato a un'identità esistente, Amazon Cognito crea una nuova identità e restituisce il nuovo ID e il token OpenID Connect per l'identità. Se l'identificatore dell'utente è già collegato, Amazon Cognito restituisce l'ID di identità preesistente e il token OpenID Connect. Memorizza gli ID di identità autenticate da sviluppatori nella cache dopo la prima richiesta e avvia le successive sessioni di base (classiche) per l'identità con `GetOpenIdTokenForDeveloperIdentity`.

La risposta a una richiesta API `GetOpenIdTokenForDeveloperIdentity` è un token generato da Amazon Cognito. Puoi inviare questo token come parametro `WebIdentityToken` in una richiesta `AssumeRoleWithWebIdentity`.

Prima di inviare il token OpenID Connect, è necessario verificarlo nell'app. Puoi utilizzare le librerie OIDC dell'SDK o una libreria come aws-jwt-verify per verificare che Amazon Cognito

abbia emesso il token. L'ID della chiave di firma, o `kid`, del token OpenID Connect è uno di quelli elencati nel documento jwks_uri† dell'identità di Amazon Cognito. Queste chiavi sono soggette a modifiche. La funzione che verifica i token di identità di Amazon Cognito deve aggiornare periodicamente l'elenco delle chiavi dal documento jwks_uri. Amazon Cognito imposta la durata dell'aggiornamento nell'intestazione della risposta `cache-control` di jwks_uri, attualmente con `max-age` impostato su 30 giorni.

Collegamento degli accessi

Come per i provider esterni, la fornitura degli accessi aggiuntivi, che non sono già associati a un'identità, implicitamente collega quegli accessi a quella identità. Se colleghi l'accesso di un provider esterno a un'identità, l'utente può utilizzare il flusso di autenticazione del provider esterno con tale provider. Tuttavia, non può usare il nome del provider degli sviluppatori nella mappa degli accessi quando chiama `GetId` o `GetOpenIdToken`.

Unione delle identità

Con le identità autenticate dagli sviluppatori, Amazon Cognito supporta sia la fusione implicita che la fusione esplicita tramite l'API. MergeDeveloperIdentities Tramite l'unione esplicita puoi contrassegnare due identità con gli identificatori utente nel tuo sistema come una singola identità. È sufficiente che tu fornisca gli identificatori dell'utente di origine e di destinazione e Amazon Cognito li unisce. La prossima volta che richiedi un token OpenID Connect per ciascuno degli identificatori dell'utente, viene restituito lo stesso ID identità.

AssumeRoleWithWebIdentity

Dopo aver ottenuto un token OpenID Connect, puoi scambiarlo con AWS credenziali temporanee tramite la richiesta AssumeRoleWithWebIdentityAPI a AWS Security Token Service ().AWS STS

Poiché non esiste alcuna restrizione sul numero di identità che puoi creare, è importante comprendere le autorizzazioni che concedi agli utenti. Configura diversi ruoli IAM per la tua applicazione: uno per gli utenti non autenticati e uno per gli utenti autenticati. La console Amazon Cognito può creare ruoli predefiniti quando configuri per la prima volta il tuo pool di identità. A questi ruoli non è effettivamente concessa alcuna autorizzazione. Modificali per soddisfare le tue esigenze.

Ulteriori informazioni su Attendibilità del ruolo e autorizzazioni.

† Il documento jwks_uri predefinito dell'identità di Amazon Cognito contiene le informazioni sulle chiavi che firmano i token per i pool di identità nella maggior parte delle Regioni AWS. Le seguenti regioni hanno documenti jwks_uri diversi.

Amazon Cognito Identity JSON web key URIs in other Regioni AWS

Regione AWS	Percorso del documento jwks_uri
AWS GovCloud (Stati Uniti occidentali)	`https://cognito-identity.us-gov-west-1.amazonaws.com/.well-known/jwks_uri`
Cina (Pechino)	`https://cognito-identity.cn-north-1.amazonaws.com.cn/.well-known/jwks_uri`
Regioni opt-in come Europa (Milano) e Africa (Città del Capo)	`https://cognito-identity.` *`Region`*`.amazonaws.com/.well-known/jwks_uri`

Puoi anche estrapolare il documento jwks_uri dall'emittente, o `iss`, che ricevi nel token OpenID da Amazon Cognito. L'endpoint di rilevamento standard OIDC `<issuer>/.well-known/openid-configuration` elenca il percorso del documento jwks_uri per il token.

Ruoli IAM

Durante la creazione di un pool di identità, ti viene chiesto di aggiornare i ruoli IAM che gli utenti assumono. I ruoli IAM funzionano in questo modo: quando un utente accede alla tua app, Amazon Cognito genera credenziali AWS temporanee per l'utente. Queste credenziali temporanee sono associate a un determinato ruolo IAM. Con il ruolo IAM, puoi definire una serie di autorizzazioni per accedere alle tue risorse. AWS

Puoi specificare ruoli IAM predefiniti per utenti autenticati e non autenticati. Inoltre, puoi definire le regole per scegliere il ruolo di ogni utente in base alle richieste nel token ID dell'utente. Per ulteriori informazioni, consulta Utilizzo del controllo degli accessi basato su ruoli.

Di default, la console Amazon Cognito crea ruoli IAM con accesso ad Amazon Cognito Sync e ad Amazon Mobile Analytics. In alternativa puoi scegliere di utilizzare ruoli IAM; esistenti.

Modificare i ruoli IAM per consentire o limitare l'accesso ad altri servizi. Per farlo, accedi alla Console IAM. Quindi seleziona Roles (Ruoli) e seleziona un ruolo. Le policy associate al ruolo selezionato sono elencate nella scheda Permissions (Autorizzazioni). Puoi personalizzare una policy di accesso

predefinita selezionando il collegamento Manage Policy (Gestisci Policy) corrispondente. Per ulteriori informazioni sull'utilizzo e per definire le policy, consulta Panoramica delle policy IAM.

> **Note**
>
> È consigliabile definire policy in grado di seguire il principio di concessione di privilegi minimi. In altre parole, le policy includono solo le autorizzazioni che gli utenti richiedono per eseguire le attività. Per ulteriori informazioni, consulta Assegnare il privilegio minimo nella Guida per l'utente di IAM.
> Ricorda che le identità non autenticate verranno assunte dagli utenti che non effettuano l'accesso alla tua app. Di solito, le autorizzazioni che assegni per le identità non autenticate devono essere più restrittive rispetto a quelle per le identità autenticate.

Argomenti

- Configurazione di una policy di attendibilità
- Policy di accesso

Configurazione di una policy di attendibilità

Amazon Cognito usa i ruoli IAM per generare credenziali temporanee per gli utenti della tua applicazione. L'accesso alle autorizzazioni è controllato da una relazione di attendibilità del ruolo. Ulteriori informazioni su Attendibilità del ruolo e autorizzazioni.

Il token presentato AWS STS viene generato da un pool di identità, che traduce un pool di utenti, un token social o di un provider OIDC o un'asserzione SAML nel proprio token. Il token del pool di identità contiene un'attestazione `aud` che è l'ID del pool di identità.

L'esempio seguente: la policy di trust dei ruoli consente al responsabile del servizio federato di chiamare l'API. `cognito-identity.amazonaws.com` AWS STS `AssumeRoleWithWebIdentity` La richiesta andrà a buon fine solo se il token del pool di identità nella richiesta API presenta le seguenti attestazioni.

1. Un'attestazione `aud` dell'ID del pool di identità `us-west-2:abcdefg-1234-5678-910a-0e8443553f95`.
2. Un'attestazione `amr` di `authenticated` viene aggiunta quando l'utente ha effettuato l'accesso e non è un utente ospite.

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

Politiche di fiducia per i ruoli IAM nell'autenticazione Basic (Classic)

È necessario applicare almeno una condizione che limiti le politiche di fiducia per i ruoli utilizzati con i pool di identità. Quando crei o aggiorni le policy di attendibilità dei ruoli per i pool di identità, IAM restituisce un errore se tenti di salvare le modifiche senza almeno una chiave di condizione che limiti le identità di origine. AWS STS non consente AssumeRoleWithWebIdentityoperazioni su più account, dai pool di identità ai ruoli IAM privi di una condizione di questo tipo.

Questo argomento include diverse condizioni che limitano le identità di origine per i pool di identità. Per un elenco completo, vedi Chiavi disponibili per la federazione delle identità AWS web.

Nell'autenticazione di base o classica con un pool di identità, puoi assumere qualsiasi ruolo IAM con AWS STS la giusta policy di fiducia. I ruoli IAM per i pool di identità di Amazon Cognito si affidano al responsabile del servizio `cognito-identity.amazonaws.com` per assumere il ruolo. Questa configurazione non è sufficiente per proteggere i ruoli IAM dall'accesso involontario alle risorse. I ruoli di questo tipo devono applicare una condizione aggiuntiva alla policy di fiducia dei ruoli. Non è possibile creare o modificare i ruoli per i pool di identità senza almeno una delle seguenti condizioni.

`cognito-identity.amazonaws.com:aud`

Limita il ruolo alle operazioni provenienti da uno o più pool di identità. Amazon Cognito indica il pool di identità di origine nel `aud` claim nel token del pool di identità.

`cognito-identity.amazonaws.com:amr`

Limita il ruolo a uno `authenticated` o agli utenti `unauthenticated` (ospiti). Amazon Cognito indica lo stato di autenticazione nel `amr` claim nel token del pool di identità.

`cognito-identity.amazonaws.com:sub`

Limita il ruolo a uno o più utenti in base all'UUID. Questo UUID è l'ID di identità dell'utente nel pool di identità. Questo valore non è il `sub` valore del provider di identità originale dell'utente. Amazon Cognito indica questo UUID nel `sub` claim del token del pool di identità.

L'autenticazione a flusso avanzato richiede che il ruolo IAM appartenga allo stesso Account AWS pool di identità, ma questo non è il caso dell'autenticazione di base.

Considerazioni aggiuntive si applicano ai pool di identità di Amazon Cognito che presuppongono ruoli IAM su più account. Le politiche di fiducia di tali ruoli devono accettare il principio del `cognito-identity.amazonaws.com` servizio e contenere la condizione specifica. `cognito-identity.amazonaws.com:aud` Per impedire l'accesso involontario alle AWS risorse, la chiave `aud` condizionale limita il ruolo agli utenti dei pool di identità indicati nel valore della condizione.

Il token emesso da un pool di identità per un'identità contiene informazioni sull'origine del pool Account AWS di identità. Quando presenti un token del pool di identità in una richiesta AssumeRoleWithWebIdentityAPI, AWS STS verifica se il pool di identità di origine ha lo stesso Account AWS ruolo di IAM. Se AWS STS determina che la richiesta è interaccount, verifica se la policy di fiducia dei ruoli presenta una `aud` condizione. La chiamata assume-role fallisce se non sono presenti tali condizioni nella policy di fiducia dei ruoli. Se la richiesta non è interaccount, AWS STS non applica questa restrizione. Come procedura consigliata, applica sempre una condizione di questo tipo alle politiche di attendibilità dei ruoli del pool di identità.

Condizioni aggiuntive relative alla politica di fiducia

Riutilizzo dei ruoli tra i pool di identità

Per riutilizzare un ruolo tra più pool di identità, poiché condividono un set di autorizzazioni, puoi includere più pool di identità come segue:

```
"StringEquals": {
    "cognito-identity.amazonaws.com:aud": [
        "us-east-1:12345678-abcd-abcd-abcd-123456790ab",
        "us-east-1:98765432-dcba-dcba-dcba-123456790ab"
    ]
}
```

Limitazione dell'accesso a specifiche identità

Per creare una policy limitata a uno specifico set di app utenti, controlla il valore di `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"
    ]
}
```

Limitazione dell'accesso a specifici provider

Per creare una policy limitata agli utenti che hanno eseguito l'accesso con un determinato provider (probabilmente il tuo provider di accesso), controlla il valore di `cognito-identity.amazonaws.com:amr`:

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "login.myprovider.myapp"
}
```

Ad esempio, un'applicazione che fa affidamento solo su Facebook avrebbe la seguente clausola amr:

```
"ForAnyValue:StringLike": {
    "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Policy di accesso

Le autorizzazioni collegate a un ruolo si applicano a tutti gli utenti che assumono tale ruolo. Per eseguire il partizionamento dell'accesso degli utenti, utilizza le condizioni e le variabili di policy. Per

ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag. Puoi utilizzare la condizione `sub` per limitare le operazioni agli ID di identità di Amazon Cognito nelle policy di accesso. Utilizza questa opzione con cautela, in particolare per identità non autenticate, prive di un ID utente coerente. Per ulteriori informazioni sulle variabili di policy IAM per la federazione web con Amazon Cognito, consulta IAM and AWS STS condition context keys nella Guida per l'AWS Identity and Access Management utente.

Per una maggiore protezione di sicurezza, Amazon Cognito applica una policy di riduzione dell'ambito alle credenziali assegnate agli utenti non autenticati nel flusso avanzato, utilizzando `GetCredentialsForIdentity`. La policy di riduzione dell'ambito aggiunge una Policy di sessione inline e una AWS politica di sessione gestita alle policy IAM applicate al ruolo non autenticato. Poiché è necessario concedere l'accesso in entrambe le policy IAM per le policy del ruolo e di sessione, la policy di riduzione dell'ambito limita l'accesso degli utenti a servizi diversi da quelli contenuti nell'elenco seguente.

> **Note**
>
> Nel flusso di base (classico), puoi creare la tua richiesta API AssumeRoleWithWebIdentity e applicare queste restrizioni alla richiesta. Come best practice di sicurezza, non assegnare autorizzazioni superiori a questa policy di riduzione dell'ambito a utenti non autenticati.

Amazon Cognito impedisce inoltre agli utenti autenticati e non autenticati di effettuare richieste API ai pool di identità di Amazon Cognito e Amazon Cognito Sync. Altri Servizi AWS potrebbero imporre restrizioni all'accesso ai servizi dalle identità web.

In caso di esito positivo della richiesta con il flusso avanzato, Amazon Cognito effettua una richiesta API `AssumeRoleWithWebIdentity` in background. Tra i parametri di questa richiesta, Amazon Cognito include quelli riportati di seguito.

1. L'ID identificativo dell'utente.
2. L'ARN del ruolo IAM che l'utente desidera assumere.
3. Un parametro `policy` che aggiunge una policy di sessione inline.
4. Un `PolicyArns.member.N` parametro il cui valore è una politica AWS gestita che concede autorizzazioni aggiuntive in Amazon. CloudWatch

Servizi cui possono accedere gli utenti non autenticati

Quando utilizzi il flusso avanzato, le policy di riduzione dell'ambito applicate da Amazon Cognito alla sessione dell'utente impediscono l'utilizzo di servizi diversi da quelli elencati nella tabella seguente. Per un sottoinsieme di servizi, sono consentite solo azioni specifiche.

Categoria	Servizio
Analisi	Amazon Data Firehose Servizio gestito da Amazon per Apache Flink
Integrazione di applicazioni	Amazon Simple Queue Service
Realtà aumentata e realtà virtuale	Amazon Sumerian
Applicazioni aziendali	Amazon Mobile Analytics Amazon Simple Email Service
Calcolo	AWS Lambda
Crittografia e PKI	AWS Key Management Service[1]
Database	Amazon DynamoDB Amazon SimpleDB
Web e dispositivi mobili front-end	AWS AppSync Servizio di posizione Amazon Amazon Simple Notification Service Amazon Pinpoint
Sviluppo dei giochi	Amazon GameLift
Internet of Things (IoT)	AWS IoT
Machine Learning	Amazon CodeWhisperer

Categoria	Servizio
	Amazon Comprehend Amazon Lex Amazon Machine Learning Amazon Personalize Amazon Polly Amazon Rekognition SageMakerAmazon[1] Amazon Textract[1] Amazon Transcribe Amazon Translate
Gestione e governance	Amazon CloudWatch CloudWatch Registri Amazon
Reti e distribuzione di contenuti	Amazon API Gateway
Sicurezza, identità e conformità	Pool di utenti Amazon Cognito
Storage	Amazon Simple Storage Service

[1] Per quanto riguarda la tabella seguente, la politica Servizi AWS in linea concede un sottoinsieme di azioni. Nella tabella sono visualizzate le azioni disponibili in ciascuno di essi.

Servizio AWS	Autorizzazioni massime per gli utenti non autenticati del flusso avanzato
AWS Key Management Service	`Encrypt` `Decrypt`

Servizio AWS	Autorizzazioni massime per gli utenti non autenticati del flusso avanzato

	`ReEncrypt`
	
	`GenerateDataKey`

Amazon SageMaker	`InvokeEndpoint`

Amazon Textract	`DetectDocumentText`
	
	`AnalyzeDocument`

Amazon Sumerian	`View*`

Per concedere l'accesso a un numero Servizi AWS superiore a questo elenco, attiva il flusso di autenticazione di base (classico) nel tuo pool di identità. Se gli utenti visualizzano errori `NotAuthorizedException` da Servizi AWS che sono consentiti dalle policy assegnate al ruolo IAM per gli utenti non autenticati, è necessario valutare se è possibile rimuovere tale servizio dal caso d'uso. Se non è possibile, passa al flusso di base.

La policy di sessione inline

La politica di sessione in linea impedisce che le autorizzazioni effettive dell'utente includano l'accesso a persone Servizi AWS esterne a quelle incluse nell'elenco seguente. È inoltre necessario concedere le autorizzazioni a questi soggetti Servizi AWS nelle politiche che si applicano al ruolo IAM dell'utente. Le autorizzazioni effettive di un utente per una sessione del ruolo assunto sono l'intersezione delle policy assegnate al relativo ruolo e policy di sessione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di AWS Identity and Access Management .

Amazon Cognito aggiunge la seguente policy inline alle sessioni per gli utenti nelle Regioni AWS abilitate per impostazione predefinita.

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

```
}
```

Per tutte le altre regioni, la policy di riduzione dell'ambito inline include tutto ciò che è elencato nelle regioni predefinite ad eccezione delle seguenti istruzioni `Action`.

```
                "cognito-sync:*",
                "sumerian:View*",
                "codewhisperer:*",
                "textract:DetectDocumentText",
                "textract:AnalyzeDocument",
                "sdb:*"
```

La politica della sessione AWS gestita

Amazon Cognito limita inoltre l'ambito delle autorizzazioni degli utenti non autenticati con la policy gestita da AWS `AmazonCognitoUnAuthedIdentitiesSessionPolicy` ai tuoi utenti non autenticati nel flusso avanzato. Questa autorizzazione deve essere specificata anche nelle policy che colleghi al ruolo IAM non autenticato.

La policy gestita `AmazonCognitoUnAuthedIdentitiesSessionPolicy` contiene le seguenti autorizzazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rum:PutRumEvents",
            "polly:*",
            "comprehend:*",
            "translate:*",
            "transcribe:*",
            "rekognition:*",
            "mobiletargeting:*",
            "firehose:*",
            "personalize:*",
            "sagemaker:InvokeEndpoint"
        ],
        "Resource": "*"
    }]
}
```

Esempi di policy di accesso

In questa sezione, sono disponibili policy di accesso di Amazon Cognito di esempio che concedono agli utenti le autorizzazioni minime necessarie per eseguire un'operazione specifica. Puoi limitare ulteriormente le autorizzazioni per un determinato ID identità utilizzando variabili di policy, laddove possibile. Ad esempio, utilizzando ${cognito-identity.amazonaws.com:sub}. Per ulteriori informazioni, consulta Informazioni sull'autenticazione di Amazon Cognito parte 3: Ruoli e policy nel blog di AWS Mobile.

> **Note**
>
> Come best practice di sicurezza, le policy devono includere solo le autorizzazioni necessarie agli utenti per eseguire le loro attività. Ciò significa che dovresti cercare sempre di definire, laddove possibile, l'ambito dell'accesso a una singola identità per gli oggetti.

Autorizzazione per l'accesso in lettura di un'identità a un singolo oggetto in Amazon S3

La seguente policy di accesso concede autorizzazioni di lettura a un'identità per recuperare un singolo oggetto da un determinato bucket S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Autorizzazione per l'accesso in lettura e scrittura di un'identità a percorsi specifici in Amazon S3

La seguente policy di accesso concede le autorizzazioni di lettura e scrittura per accedere a una specifica "cartella" prefisso in un bucket S3 mappando il prefisso alla variabile `${cognito-identity.amazonaws.com:sub}`.

Con questa policy, un'identità come `us-east-1:12345678-1234-1234-1234-123456790ab` inserita tramite `${cognito-identity.amazonaws.com:sub}` sarà in grado di ottenere, inserire ed elencare oggetti in `arn:aws:s3:::mybucket/us-east-1:12345678-1234-1234-1234-123456790ab`. Tuttavia, all'identità non verrà concesso l'accesso ad altri oggetti in `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}/*"]
    }
  ]
}
```

Assegnazione dell'accesso granulare delle identità ad Amazon DynamoDB

La seguente policy di accesso fornisce un controllo granulare dell'accesso alle risorse DynamoDB utilizzando variabili d'ambiente Amazon Cognito. Tali variabili concedono l'accesso agli elementi in DynamoDB attraverso l'ID identità. Per ulteriori informazioni, consulta Utilizzo di condizioni delle policy IAM per il controllo granulare degli accessi nella Guida per lo sviluppatore di 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}"]
        }
      }
    }
  ]
}
```

Autorizzazione di un'identità a richiamare una funzione Lambda

La seguente policy di accesso concede a un'identità l'autorizzazione per richiamare una funzione Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
             "Effect": "Allow",
             "Action": "lambda:InvokeFunction",
             "Resource": [
                 "arn:aws:lambda:us-west-2:123456789012:function:MyFunction"
             ]
         }
     ]
 }
```

Autorizzazione di un'identità a pubblicare record in flussi di dati Kinesis

La policy di accesso seguente consente a un'identità di utilizzare l'operazione `PutRecord` con qualsiasi Kinesis Data Streams. Può essere applicata agli utenti che devono aggiungere record di

dati a tutti i flussi in un account. Per ulteriori informazioni, consulta Controllo dell'accesso alle risorse Amazon Kinesis Data Streams che utilizzano IAM nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:PutRecord",
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/stream1"
            ]
        }
    ]
}
```

Autorizzazione di un'identità ad accedere ai propri dati nell'archivio di Amazon Cognito Sync

La seguente policy di accesso concede a un'identità le autorizzazioni solo per i dati nell'archivio di 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}/*"]
      }]
  }
```

Attendibilità del ruolo e autorizzazioni

Questi ruoli differiscono nelle loro relazioni di attendibilità. Di seguito viene riportato un esempio di policy di attendibilità per i ruoli non autenticati:

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

Questa policy concede l'autorizzazione agli utenti federati di `cognito-identity.amazonaws.com` (approvatore del token OpenID Connect) di assumere questo ruolo. Inoltre la policy limita l'`aud` del token, in questo caso l'ID dei pool di identità, in base al pool di identità. Infine, la policy specifica che uno dei membri dell'array dell'attestazione `amr` con più valori del token emesso dall'operazione API `GetOpenIdToken` Amazon Cognito ha il valore `unauthenticated`.

Quando Amazon Cognito crea un token, imposta l'`amr` del token o come `unauthenticated` o `authenticated`. Se `amr` è `authenticated`, il token include tutti i provider utilizzati durante l'autenticazione. In questo modo, puoi creare un ruolo che fa affidamento solo sugli utenti che eseguono l'accesso tramite Facebook. Ti basterà cambiare la condizione `amr` in modo che sia somigliante alla seguente:

```
"ForAnyValue:StringLike": {
  "cognito-identity.amazonaws.com:amr": "graph.facebook.com"
}
```

Presta attenzione quando vengono modificate le relazioni di affidabilità sui tuoi ruoli, oppure quando si tenta di utilizzare ruoli tra pool di identità. Se il ruolo non è configurato per fare affidamento in maniera corretta sul tuo pool di identità, STS genera un'eccezione come la seguente:

```
AccessDenied -- Not authorized to perform sts:AssumeRoleWithWebIdentity
```

Se vedi questo mesaggio, verifica che il tuo pool di identità e il tipo di autenticazione siano appropriati per il ruolo.

Best practice di sicurezza per i pool di identità di Amazon Cognito

I pool di identità di Amazon Cognito forniscono AWS credenziali temporanee per la tua applicazione. Account AWS spesso contengono sia le risorse di cui gli utenti dell'applicazione hanno bisogno, sia risorse private di back-end. I ruoli e le policy IAM che costituiscono le AWS credenziali possono concedere l'accesso a qualsiasi di queste risorse.

La best practice principale per la configurazione del pool di identità consiste nel garantire che l'applicazione possa svolgere il lavoro senza privilegi eccessivi o involontari. Per evitare errori di configurazione della sicurezza, consulta questi consigli prima del lancio di ogni applicazione che desideri rilasciare in produzione.

Argomenti

- Le migliori pratiche di configurazione IAM
- Best practice per la configurazione del pool di identità

Le migliori pratiche di configurazione IAM

Quando un ospite o un utente autenticato avvia una sessione nell'applicazione che richiede le credenziali del pool di identità, l'applicazione recupera le AWS credenziali temporanee per un ruolo IAM. Le credenziali potrebbero riguardare un ruolo predefinito, un ruolo scelto dalle regole nella configurazione del pool di identità o un ruolo personalizzato scelto dall'app. Con le autorizzazioni assegnate a ciascun ruolo, l'utente ottiene l'accesso alle risorse. AWS

Per ulteriori informazioni sulle best practice generali di IAM, consulta le best practice di IAM nella Guida per l' AWS Identity and Access Management utente.

Utilizza le condizioni delle policy di fiducia nei ruoli IAM

IAM richiede che i ruoli per i pool di identità abbiano almeno una condizione di policy di fiducia. Questa condizione può, ad esempio, impostare l'ambito del ruolo solo per gli utenti autenticati. AWS STS richiede inoltre che le richieste di autenticazione di base tra account abbiano due condizioni specifiche: `cognito-identity.amazonaws.com:aud` e. `cognito-identity.amazonaws.com:amr` Come best practice, applica entrambe queste condizioni in tutti i

ruoli IAM che si affidano al principale `cognito-identity.amazonaws.com` del servizio di pool di identità.

- `cognito-identity.amazonaws.com:aud`: L'affermazione aud nel token del pool di identità deve corrispondere a un ID affidabile del pool di identità.
- `cognito-identity.amazonaws.com:amr`: L'attestazione amr nel token del pool di identità deve essere autenticata o non autenticata. Con questa condizione, puoi riservare l'accesso a un ruolo solo agli ospiti non autenticati o solo agli utenti autenticati. È possibile rifinire ulteriormente il valore di questa condizione per limitare il ruolo agli utenti di un provider specifico, ad esempio. `graph.facebook.com`

Il seguente esempio di politica di fiducia dei ruoli concede l'accesso a un ruolo nelle seguenti condizioni:

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

Elementi che si riferiscono ai pool di identità

- `"Federated": "cognito-identity.amazonaws.com"`: gli utenti devono provenire da un pool di identità.
- `"cognito-identity.amazonaws.com:aud": "us-east-1:a1b2c3d4-5678-90ab-cdef-example11111"`: Gli utenti devono provenire dal pool di identità specificous-east-1:a1b2c3d4-5678-90ab-cdef-example11111.
- `"cognito-identity.amazonaws.com:amr": "authenticated"`: Gli utenti devono essere autenticati. Gli utenti ospiti non possono assumere il ruolo.

Applica le autorizzazioni con privilegi minimi

Quando imposti le autorizzazioni con le policy IAM per l'accesso autenticato o l'accesso ospite, concedi solo le autorizzazioni specifiche necessarie per eseguire attività specifiche o le autorizzazioni con privilegi minimi. La seguente policy IAM di esempio, se applicata a un ruolo, concede l'accesso in sola lettura a un singolo file di immagine in un bucket Amazon S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::mybucket/assets/my_picture.jpg"]
    }
  ]
}
```

Best practice per la configurazione del pool di identità

I pool di identità offrono opzioni flessibili per la generazione di AWS credenziali. Non utilizzare scorciatoie di progettazione quando l'applicazione può funzionare con i metodi più sicuri.

Comprendi gli effetti dell'accesso degli ospiti

L'accesso non autenticato come ospite consente agli utenti di recuperare i dati da te Account AWS prima di effettuare l'accesso. Chiunque conosca l'ID del tuo pool di identità può richiedere credenziali

non autenticate. L'ID del tuo pool di identità non è un'informazione riservata. Quando attivi l'accesso come ospite, le AWS autorizzazioni concesse alle sessioni non autenticate sono disponibili per tutti.

Come procedura consigliata, lascia disattivato l'accesso degli ospiti e recupera le risorse necessarie solo dopo l'autenticazione degli utenti. Se l'applicazione richiede l'accesso alle risorse prima dell'accesso, prendi le seguenti precauzioni.

- Acquisite familiarità con le limitazioni automatiche imposte ai ruoli non autenticati.
- Monitora e modifica le autorizzazioni dei ruoli IAM non autenticati per soddisfare le esigenze specifiche della tua applicazione.
- Concedi l'accesso a risorse specifiche.
- Proteggi la policy di fiducia del tuo ruolo IAM predefinito non autenticato.
- Attiva l'accesso come ospite solo quando sei sicuro di poter concedere le autorizzazioni nel tuo ruolo IAM a chiunque su Internet.

Utilizza l'autenticazione avanzata per impostazione predefinita

Con l'autenticazione di base (classica), Amazon Cognito delega la selezione del ruolo IAM alla tua app. Al contrario, il flusso avanzato utilizza la logica centralizzata nel pool di identità per determinare il ruolo IAM. Fornisce inoltre una sicurezza aggiuntiva per le identità non autenticate con una politica mirata che stabilisce un limite massimo per le autorizzazioni IAM. Il flusso avanzato è la scelta più sicura con il livello più basso di impegno da parte degli sviluppatori. Per ulteriori informazioni su queste opzioni, consultaFlusso di autenticazione dei pool di identità (identità federate).

Il flusso di base può esporre la logica lato client che riguarda la selezione dei ruoli e l'assemblaggio della richiesta di credenziali dell'API AWS STS. Il flusso migliorato nasconde sia la logica che la richiesta di assunzione del ruolo alla base dell'automazione del pool di identità.

Quando configuri l'autenticazione di base, applica le migliori pratiche IAM ai tuoi ruoli IAM e alle relative autorizzazioni.

Utilizza i provider di sviluppo in modo sicuro

Le identità autenticate dagli sviluppatori sono una funzionalità dei pool di identità per le applicazioni lato server. L'unica prova di autenticazione richiesta dai pool di identità per l'autenticazione degli sviluppatori sono le AWS credenziali di uno sviluppatore di pool di identità. I pool di identità non impongono alcuna restrizione sulla validità degli identificatori sviluppatore-fornitore presenti in questo flusso di autenticazione.

Come best practice, implementa i provider di sviluppo solo nelle seguenti condizioni:

- Per creare la responsabilità per l'uso di credenziali autenticate dagli sviluppatori, create il nome e gli identificatori del provider di sviluppo in modo da indicare la fonte di autenticazione. Ad esempio: `"Logins" : {"MyCorp provider" : "[provider application ID]"}`.
- Evita le credenziali utente di lunga durata. Configura il tuo client lato server per richiedere identità con ruoli collegati ai servizi come i profili di istanza EC2 e i ruoli di esecuzione Lambda.
- Evita di mescolare fonti di fiducia interne ed esterne nello stesso pool di identità. Aggiungi il tuo provider di sviluppo e i provider Single Sign-On (SSO) in pool di identità separati.

Utilizzo di attributi per il controllo degli accessi

Gli attributi per il controllo degli accessi è l'implementazione dei pool di identità di Amazon Cognito del controllo degli accessi basato su attributi (ABAC). È possibile utilizzare le policy IAM per controllare l'accesso alle risorse AWS attraverso i pool di identità di Amazon Cognito in base agli attributi dell'utente. Questi attributi possono essere ricavati da provider di identità social e aziendali. È possibile eseguire la mappatura degli attributi all'interno dei token di accesso e ID dei provider o delle asserzioni SAML per i tag a cui è possibile fare riferimento nelle policy di autorizzazione IAM.

Puoi scegliere le mappature predefinite oppure creare le tue mappature personalizzate nei pool di identità di Amazon Cognito. Le mappature predefinite consentono di scrivere policy IAM in base a un set fisso di attributi utente. Le mappature personalizzate consentono invece di selezionare un set personalizzato di attributi utente a cui si fa riferimento nelle policy di autorizzazioni IAM. I nomi degli attributi nella console Amazon Cognito sono mappati alla chiave tag per principal, che sono i tag a cui si fa riferimento nella policy di autorizzazioni IAM.

Ad esempio, supponiamo che tu sia proprietario di un servizio di streaming multimediale con un abbonamento gratuito e uno pagamento. Puoi archiviare i file multimediali in Amazon S3 e taggarli con tag gratuiti o premium. Puoi utilizzare gli attributi per il controllo dell'accesso per consentire l'accesso a contenuti gratuiti e a pagamento in base al livello di appartenenza dell'utente, che è parte del profilo dell'utente. È possibile eseguire la mappatura dell'attributo di appartenenza per una chiave di tag per il principal da passare alla policy di autorizzazioni IAM. In questo modo puoi creare una singola policy di autorizzazioni e consentire in modo condizionale l'accesso al contenuto premium in base al valore del livello di appartenenza e del tag nei file di contenuto.

Argomenti

- Utilizzo degli attributi per il controllo dell'accesso con i pool di identità di Amazon Cognito

- Esempio di utilizzo di attributi per la policy di controllo degli accessi
- Disattivazione di attributi per il controllo degli accessi (console)
- Mappature di provider predefinite

L'uso degli attributi per controllare l'accesso offre molti vantaggi:

- La gestione delle autorizzazioni è molto più efficiente quando si utilizzano gli attributi per il controllo degli accessi. È possibile creare una policy di autorizzazioni di base che utilizzi attributi utente anziché creare più policy per funzioni di lavori diversi.
- Non è necessario aggiornare le policy ogni volta che si aggiungono o rimuovono risorse o utenti per l'applicazione. La policy di autorizzazione concede l'accesso solo agli utenti con gli attributi utente corrispondenti. Ad esempio, potrebbe essere necessario controllare l'accesso a determinati bucket S3 in base alla mansione degli utenti. In tal caso, puoi creare una policy di autorizzazioni per consentire l'accesso a questi file solo per gli utenti all'interno della mansione definita. Per ulteriori informazioni, consulta Tutorial IAM: Utilizzo dei tag di sessione SAML per ABAC.
- Gli attributi possono essere passati come tag principali a una policy che consente o nega le autorizzazioni in base ai valori di tali attributi.

Utilizzo degli attributi per il controllo dell'accesso con i pool di identità di Amazon Cognito

Prima di utilizzare gli attributi per il controllo degli accessi, assicurati di soddisfare i seguenti prerequisiti:

- Un account AWS
- Pool di utenti
- Pool di identità
- Configurazione di un SDK
- Provider di identità integrati
- Credenziali

Per utilizzare attributi per il controllo degli accessi, la richiesta impostata come origine del set di dati imposta il valore della Chiave tag scelta. Amazon Cognito applica la chiave e il valore del tag alla sessione dell'utente. Le policy IAM possono valutare l'accesso dell'utente dalla

condizione `${aws:PrincipalTag/tagkey}`. IAM valuta il valore del tag dell'utente rispetto alla policy.

È necessario preparare i ruoli IAM le cui credenziali si desidera passare agli utenti. La policy di attendibilità di questi ruoli deve consentire ad Amazon Cognito di assumere il ruolo per l'utente. Per quanto riguarda gli attributi per il controllo degli accessi, devi inoltre consentire ad Amazon Cognito di applicare i tag principali alla sessione temporanea dell'utente. Concedi l'autorizzazione per assumere il ruolo con l'azione AssumeRoleWithWebIdentity. Concedi l'autorizzazione per applicare tag alle sessioni degli utenti con l'azione basata solo sull'autorizzazione `sts:TagSession`. Per ulteriori informazioni, consulta Passare i tag di sessione in AWS Security Token Service nella Guida per l'utente di AWS Identity and Access Management. Per una policy di attendibilità che concede le autorizzazioni `sts:AssumeRoleWithWebIdentity` e `sts:TagSession` al principale del servizio Amazon Cognito`cognito-identity.amazonaws.com`, consulta Esempio di utilizzo di attributi per la policy di controllo degli accessi.

Per configurare gli attributi per il controllo degli accessi nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare. Se desideri aggiungere un nuovo IdP, seleziona Aggiungi provider di identità.
4. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, scegli Modifica in Attributi per il controllo degli accessi.
    a. Per non applicare alcun tag principale, scegli Inattivo.
    b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
    c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.
5. Selezionare Save changes (Salva modifiche).

Esempio di utilizzo di attributi per la policy di controllo degli accessi

Consideriamo uno scenario in cui un dipendente dell'ufficio legale di un'azienda deve riportare nei bucket tutti i file che appartengono al suo reparto e classificarli con il loro livello di sicurezza. Si supponga che il token che questo dipendente ottiene dal provider di identità contenga le seguenti attestazioni.

Attestazioni

```
{ .
  .
"sub" : "57e7b692-4f66-480d-98b8-45a6729b4c88",
"department" : "legal",
"clearance" : "confidential",
 .
 .
}
```

È possibile eseguire la mappatura di questi attributi per i tag e si può fare riferimento ad essi nelle policy di autorizzazioni IAM come tag principal. Adesso puoi possibile gestire l'accesso modificando il profilo utente sul provider di identità. In alternativa, puoi modificare gli attributi sul lato risorsa utilizzando nomi o tag senza modificare la policy stessa.

La seguente policy di autorizzazioni esegue due operazioni:

- Consente l'accesso all'elenco a tutti i bucket S3 che terminano con un prefisso che corrisponde al nome del reparto dell'utente.
- Consente l'accesso in lettura ai file in questi bucket, purché il tag di autorizzazione sul file corrisponda all'attributo di autorizzazione dell'utente.

Policy delle autorizzazioni

```
{
 "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 policy di attendibilità determina chi può assumere questo ruolo. La policy di relazione di attendibilità consente l'utilizzo di `sts:AssumeRoleWithWebIdentity` e `sts:TagSession` per consentire l'accesso. Aggiunge condizioni per limitare la policy al pool di identità creato e garantisce che sia per un ruolo autenticato.

Policy di trust

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

Disattivazione di attributi per il controllo degli accessi (console)

Segui questa procedura per disattivare gli attributi per il controllo degli accessi.

Per disattivare gli attributi per il controllo degli accessi nella console

1. Accedi alla console di Amazon Cognito e seleziona Pool di identità. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Individua Provider di identità. Scegli il provider di identità da modificare.
4. Scegli Modifica in Attributi per il controllo degli accessi.
5. Per non applicare alcun tag principale, scegli Inattivo.
6. Selezionare Save changes (Salva modifiche).

Mappature di provider predefinite

Nella tabella seguente sono riportate le informazioni di mappatura predefinite per i provider di autenticazione supportati da Amazon Cognito.

Provider	Tipo di token	Valori dei tag principal	Esempio
Bacino d'utenza di Amazon Cognito	Token ID	aud(client ID) and sub(user ID)	"6jk8ltokc7ac9es6jrtg9q572f", "57e7b692-4f66-480d-98b8-45a6729b4c88"
Facebook	Token di accesso	aud(app_id), sub(user_id)	"492844718097981", "112177216992379"
Google	Token ID	aud(client ID) and sub(user ID)	"620493171733-eebk7c0hcp5lj3e1tlqp1g

Provider	Tipo di token	Valori dei tag principal	Esempio
			ntt3k0rncv.apps.go ogleusercontent.com", "10922006345240474 6097"
SAML	Asserzioni	"http://schemas.xm lsoap.org/ws/2005/ 05/identity/claims /nameidentifier" , "http://schemas.xm lsoap.org/ws/2005/05/ identity/claims/name"	"auth0|5e28d196f8f 55a0eaaa95de3", "user123@gmail.com"
Apple	Token ID	aud(client ID) and sub (user ID)	"com.amazonaws.ec2 -54-80-172-243.com pute-1.client", "001968.a6ca34e9c1 e742458a26cf800585 4be9.0733"
Amazon	Token di accesso	aud (Client ID on Amzn Dev Ac), user_id(user ID)	"amzn1.application -oa2-client.9d70d9 382d34461 08aaee3dd763a0fa6", "amzn1.account.AGH NIFJQMFSB G3G6XCPVB 35ORQAA"
Provider OIDC standard	Token ID e token di accesso	aud (as client_id), sub (as user ID)	"620493171733-eebk 7c0hcp5lj3e1tlqp1g ntt3k0rncv.apps.go ogleusercontent.com", "10922006345240474 6097"

Provider	Tipo di token	Valori dei tag principal	Esempio
Twitter	Token di accesso	aud (app ID; app Secret), sub (user ID)	"DfwifTtKEX1FiIBRnOTIR0CFK; Xgj5xb8xIrIVCPjXgLIdkW7fXmwcJJrFvnoK9gwZkLexo1y5z1", "1269003884292222976"
DevAuth	Eseguire la mappatura	Non applicabile	"tag1", "tag2"

> **Note**
>
> L'opzione di mappatura degli attributi predefinita viene popolata automaticamente per Chiave tag per principal e Nomi attributo. Non è possibile modificare le mappature predefinite.

Utilizzo del controllo degli accessi basato su ruoli

I pool di identità di Amazon Cognito assegnano agli utenti autenticati una serie di credenziali temporanee con privilegi limitati per accedere alle tue risorse. AWS Le autorizzazioni di ciascun utente vengono controllate attraverso i ruoli IAM che hai creato. Puoi definire le regole per scegliere il ruolo di ogni utente sulla base delle richieste nel token ID dell'utente. Puoi definire un ruolo di default per gli utenti autenticati. Puoi inoltre definire un ruolo IAM separato con autorizzazioni limitate per gli utenti guest non autenticati.

Creazione dei ruoli per la mappatura dei ruoli

È importante aggiungere la policy di attendibilità appropriata per ogni ruolo, in modo che possa essere assunta da Amazon Cognito solo per gli utenti autenticati nel pool di identità. Di seguito viene riportato un esempio di suddetta policy di attendibilità:

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

Questa policy consente agli utenti federati di `cognito-identity.amazonaws.com` (approvatore del token OpenID Connect) di assumere questo ruolo. Inoltre la policy limita l'`aud` del token, in questo caso l'ID dei pool di identità, in base al pool di identità. Infine, la policy specifica che uno dei membri dell'array dell'attestazione `amr` con più valori del token emesso dall'operazione API `GetOpenIdToken` Amazon Cognito ha il valore `authenticated`.

Concessione delle autorizzazioni per il passaggio di ruoli

Per consentire a un utente di impostare ruoli con autorizzazioni superiori rispetto alle autorizzazioni dell'utente esistenti su un pool di identità, concedere loro l'autorizzazione `iam:PassRole` per trasferire il ruolo all'API `set-identity-pool-roles`. Ad esempio, se l'utente non è in grado di scrivere su Amazon S3 ma il ruolo IAM che l'utente imposta nel pool di identità concede l'autorizzazione per scrivere su Amazon S3, l'utente può impostare questo ruolo solo se a quest'ultimo viene concessa l'autorizzazione `iam:PassRole`. Il seguente esempio di policy mostra come concedere l'autorizzazione `iam:PassRole`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::123456789012:role/myS3WriteAccessRole"
            ]
        }
    ]
}
```

In questo esempio di policy viene concessa l'autorizzazione `iam:PassRole` al ruolo `myS3WriteAccessRole`. Il ruolo viene specificato usando l'Amazon Resource Name (ARN) del ruolo. È necessario, inoltre, collegare questa policy all'utente. Per ulteriori informazioni, consulta la sezione relativa all'utilizzo di policy gestite.

> **Note**
>
> Le funzioni Lambda usano una policy basata sulle risorse collegata direttamente alla funzione Lambda stessa. Quando crei una regola che invoca una funzione Lambda, non trasferisci un ruolo; pertanto, l'utente che crea la regola non richiede l'autorizzazione `iam:PassRole`. Per ulteriori informazioni sull'autorizzazione della funzione Lambda, consulta Gestione delle autorizzazioni: uso di una policy della funzione Lambda.

Utilizzo dei token per l'assegnazione dei ruoli agli utenti

Per gli utenti che effettuano l'accesso tramite il bacino d'utenza Amazon Cognito, i ruoli possono essere trasferiti nel token ID assegnato dal bacino d'utenza. I ruoli vengono visualizzati nelle seguenti attestazioni nel token ID:

- L'attestazione `cognito:preferred_role` è il ruolo ARN.
- L'attestazione `cognito:roles` è una stringa separata da virgole contenente una serie di ARN del ruolo consentite.

Le attestazioni sono impostate come segue:

- L'attestazione `cognito:preferred_role` è impostata sul ruolo del gruppo con il valore `Precedence` migliore (più basso). Se viene consentito un solo ruolo,

`cognito:preferred_role` è impostato su tale ruolo. Se sono presenti più ruoli e nessun ruolo singolo ha la migliore precedenza, l'attestazione non è stata impostata.

- L'attestazione `cognito:roles` è impostata se è presente almeno un ruolo.

Quando si utilizzano i token per l'assegnazione dei ruoli, se sono presenti più ruoli che possono essere assegnati all'utente, i pool di identità di Amazon Cognito (identità federate) selezionano il ruolo come segue:

- Utilizza il GetCredentialsForIdentity`CustomRoleArn`parametro se è impostato e corrisponde a un ruolo nel claim. `cognito:roles` Se questo parametro non corrisponde a un ruolo in `cognito:roles`, nega l'accesso.
- Se l'attestazione `cognito:preferred_role` è impostata, utilizzala.
- Se l'`cognito:preferred_role`attestazione non è impostata, viene impostata e non `CustomRoleArn` è specificata nella chiamata a`GetCredentialsForIdentity`, quindi l'impostazione della risoluzione del ruolo nella console o nel `AmbiguousRoleResolution` campo (nel `RoleMappings` parametro dell'SetIdentityPoolRolesAPI) viene utilizzata per determinare il ruolo da assegnare. `cognito:roles`

Utilizzo di una mappatura basata su regole per assegnare i ruoli agli utenti

Le regole ti permettono di eseguire la mappatura delle attestazioni del token di un provider di identità per i ruoli IAM.

Ogni regola specifica l'attestazione di un token (ad esempio un attributo utente nel token ID di un bacino d'utenza di Amazon Cognito), il tipo di corrispondenza, un valore e un ruolo IAM. Il tipo di corrispondenza può essere `Equals`, `NotEqual`, `StartsWith`, o `Contains`. Se un utente dispone di un valore corrispondente all'attestazione, tale utente può assumere quel ruolo nel momento in cui ottiene le credenziali. Ad esempio, puoi creare una regola che assegna un determinato ruolo IAM agli utenti con un valore di attributo personalizzato `custom:dept` di `Sales`.

> **Note**
>
> Nelle impostazioni delle regole, gli attributi personalizzati richiedono il prefisso `custom:` per distinguersi dagli attributi standard.

Le regole vengono valutate in ordine e viene utilizzato il ruolo IAM per la prima regola corrispondente, a meno che non sia specificato `CustomRoleArn` per sovrascrivere l'ordine. Per ulteriori informazioni sugli attributi utente nei bacini d'utenza di Amazon Cognito, consulta Attributi del bacino d'utenza.

Puoi impostare più regole per un provider di autenticazione nella console del pool di identità (identità federate). Le regole vengono applicate in ordine. Puoi trascinare le regole per modificarne l'ordine. La prima regola corrispondente ha la priorità. Se il tipo di corrispondenza è `NotEqual` e l'attestazione non esiste, la regola non viene valutata. Se nessuna delle regole corrisponde, l'impostazione Risoluzione del ruolo viene applicata su Utilizza ruolo autenticato predefinito o Nega richiesta.

Nell'API e nella CLI, puoi specificare il ruolo da assegnare quando nessuna regola corrisponde nel `AmbiguousRoleResolution` campo del RoleMappingtipo, che è specificato nel `RoleMappings` parametro dell'SetIdentityPoolRolesAPI.

Puoi configurare la mappatura basata su regole per i provider di identità OpenID Connect (OIDC) e SAML nell'API o con AWS CLI il campo del tipo. `RulesConfiguration` RoleMapping È possibile specificare questo campo nel parametro dell'API. `RoleMappings` SetIdentityPoolRoles Al AWS Management Console momento non consente di aggiungere regole per i provider OIDC o SAML.

Ad esempio, il AWS CLI comando seguente aggiunge una regola che assegna il ruolo `arn:aws:iam::123456789012:role/Sacramento_team_S3_admin` agli utenti nella tua sede di Sacramento che sono stati autenticati da 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
```

Contenuto di **`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"
                    }
                ]
            }
        }
    }
}
```

Per ogni bacino d'utenza o un altro provider di autenticazione configurato per un pool di identità, puoi creare fino a 25 regole. Questo limite non è regolabile. Per ulteriori informazioni, consulta la sezione Quote in Amazon Cognito.

Attestazioni dei token da utilizzare nella mappatura basata su regole

Amazon Cognito

Un token ID di Amazon Cognito è rappresentato come un token Web JSON (JWT). Il token contiene attestazioni relative all'identità dell'utente autenticato, ad esempio `name` `family_name`, e `phone_number`. Per ulteriori informazioni sulle attestazioni standard, consulta la specifica OpenID Connect. Oltre alle attestazioni standard, di seguito sono riportate le attestazioni aggiuntive specifiche di Amazon Cognito:

- `cognito:groups`
- `cognito:roles`
- `cognito:preferred_role`

Amazon

Le seguenti attestazioni, insieme ai valori possibili per dette attestazioni, possono essere utilizzate con Login with Amazon:

- `iss`: www.amazon.com
- `aud`: Id app
- `sub`: `sub` dal token di Login with Amazon

Facebook

Le seguenti attestazioni, insieme ai valori possibili per dette attestazioni, possono essere utilizzate con Facebook:

- `iss`: graph.facebook.com
- `aud`: Id app
- `sub`: `sub` dal token di Facebook

Google

Un token di Google contiene attestazioni standard della specifica OpenID Connect. Tutte le attestazioni nel token di OpenID sono disponibili per la mappatura basata su regole. Consulta il sito OpenID Connect di Google per scoprire quali attestazioni sono disponibili nel token di Google.

Apple

Un token Apple contiene attestazioni standard della specifica OpenID Connect. Consulta Authenticating Users with Sign in with Apple (Autenticazione degli utenti con Accedi con Apple) nella documentazione di Apple per ulteriori informazioni sulle attestazioni disponibili tramite token Apple. Il token di Apple non sempre contiene `email`.

OpenID

Tutte le attestazioni nel token di Open Id sono disponibili per la mappatura basata su regole. Per ulteriori informazioni sulle attestazioni standard, consulta la specifica OpenID Connect. Consulta la documentazione relativa al provider di OpenID per scoprire le altre attestazioni aggiuntive disponibili.

SAML

Le attestazioni vengono ricavate analizzando l'asserzione SAML ricevuta. Tutte le attestazioni disponibili nell'asserzione SAML possono essere utilizzate nella mappatura basata su regole.

Best practice per il controllo accessi basato sui ruoli

> **Important**
>
> Se l'attestazione che si sta mappando a un ruolo può essere modificata dall'utente finale, qualsiasi utente finale può assumere il ruolo e impostare la policy di conseguenza. Esegui la

mappatura solo delle attestazioni che non possono essere direttamente impostate dall'utente finale ai ruoli con autorizzazioni elevate. In un bacino d'utenza di Amazon Cognito puoi impostare le autorizzazioni di lettura e di scrittura per app per ogni attributo utente.

Important

Se imposti i ruoli per i gruppi in un bacino d'utenza di Amazon Cognito, tali ruoli vengono trasmessi tramite il token ID dell'utente. Per utilizzare questi ruoli, è necessario impostare anche Choose role from token (Scegli ruolo dal token) per la selezione del ruolo autenticato per il pool di identità.
È possibile utilizzare l'impostazione della risoluzione dei ruoli nella console e il `RoleMappings` parametro dell'SetIdentityPoolRolesAPI per specificare qual è il comportamento predefinito quando non è possibile determinare il ruolo corretto dal token.

Ottenere le credenziali

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS In questa sezione viene descritto come ottenere le credenziali e come recuperare un'identità di Amazon Cognito da un pool di identità.

Amazon Cognito supporta sia le identità autenticate che le identità non autenticate. L'identità degli utenti non autenticati non è verificata. Ciò rende questo ruolo appropriato per utenti guest dell'app o per i casi in cui non importa se l'identità degli utenti è verificata. Gli utenti autenticati accedono all'applicazione tramite un provider di identità di terza parte o un bacino d'utenza, che verifica la loro identità. Assicurati di creare l'ambito delle autorizzazioni di risorse in modo appropriato per evitare che utenti non autenticati possano accedere a esse.

Le identità di Amazon Cognito non sono credenziali. Vengono scambiate con credenziali utilizzando il supporto per la federazione delle identità Web in (). AWS Security Token Service AWS STS Per ottenere le credenziali AWS per gli utenti dell'app, ti consigliamo di utilizzare `AWS.CognitoIdentityCredentials`. L'identità nell'oggetto credentials viene quindi scambiata con credenziali utilizzando. AWS STS

> **Note**
>
> Se hai creato il pool di identità prima di febbraio 2015, devi riassociare i ruoli con il pool di identità per utilizzare il costruttore `AWS.CognitoIdentityCredentials` senza i ruoli come parametri. A tale scopo, apri console Amazon Cognito, scegli Manage identity pools (Gestisci pool di identità), seleziona il pool di identità, scegli Edit identity Pool (Modifica pool di identità), specifica i ruoli autenticati e non autenticati e salva le modifiche.

I provider di credenziali di identità web fanno parte della catena di provider di credenziali predefinita negli SDK AWS . Per impostare il token del pool di identità in un `config` file locale per un AWS SDK o il AWS CLI, aggiungi una voce di profilo. `web_identity_token_file` Consulta Assume il ruolo di fornitore di credenziali nella Guida di riferimento agli AWS SDK e agli strumenti.

Per ulteriori informazioni su come compilare le credenziali di identità web nell'SDK, fai riferimento alla guida per gli sviluppatori di SDK. Per ottenere i migliori risultati, inizia il tuo progetto con l'integrazione del pool di identità integrata in. AWS Amplify

AWS Risorse SDK per ottenere e impostare credenziali con pool di identità

- Federazione del pool di identità (Android) in Amplify Dev Center
- Federazione del pool di identità (iOS) in Amplify Dev Center
- Utilizzo di Amazon Cognito Identity per autenticare gli utenti nella Developer Guide AWS SDK for JavaScript
- Provider di credenziali Amazon Cognito nella Developer Guide AWS SDK for .NET
- Specificate le credenziali in modo programmatico nella Guida per gli sviluppatori AWS SDK for Go
- Fornisci credenziali temporanee nel codice contenuto nella Guida per gli sviluppatori AWS SDK for Java 2.x
- assumeRoleWithWebIdentityCredentialProviderprovider nella Guida per gli AWS SDK for PHP sviluppatori
- Assumere il ruolo con provider di identità web nella documentazione AWS SDK for Python (Boto3)
- Specificazione delle credenziali e della regione predefinita nella Guida per gli sviluppatori AWS SDK for Rust

Le sezioni seguenti forniscono esempi di codice in alcuni SDK precedenti AWS .

Android

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui i passaggi seguenti.

Per utilizzare un pool di identità di Amazon Cognito in un'app Android, configura. AWS Amplify Per ulteriori informazioni, consultare Autenticazione in Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore univoco di Amazon Cognito (ID identità) per l'utente finale. Se esegui l'autenticazione degli utenti, puoi recuperare l'ID identità dopo aver impostato i token di accesso nel provider di credenziali:

```
String identityId = credentialsProvider.getIdentityId();
Log.d("LogTag", "my ID is " + identityId);
```

> **Note**
>
> Non chiamare `getIdentityId()`, `refresh()` o `getCredentials()` nel thread principale dell'applicazione. A partire da Android 3.0 (API Level 11), l'app fallirà automaticamente e genererà un errore NetworkOnMainThreadExceptionse esegui I/O di rete sul thread principale dell'applicazione. È necessario spostare il codice in un thread in background utilizzando `AsyncTask`. Per ulteriori informazioni, consulta la documentazione di Android. Puoi inoltre chiamare `getCachedIdentityId()` per recuperare un ID, ma solo se ce n'è già uno memorizzato nella cache in locale. In caso contrario, il metodo restituirà un valore nullo.

iOS - Objective-C

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS I pool di identità di Amazon Cognito supportano sia le identità autenticate che le identità non autenticate. Per fornire AWS le credenziali alla tua app, completa i seguenti passaggi.

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication e Flutter Authentication nella Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale se autorizzi gli utenti non autenticati o dopo aver impostato i token di accesso nel fornitore di credenziali se autentichi gli utenti:

```
// 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` è una chiamata asincrona. Se sul tuo provider è già impostato un ID identità, puoi chiamare `credentialsProvider.identityId` per recuperare tale identità, la quale è memorizzata nella cache in locale. Tuttavia, se sul provider non è impostato alcun ID identità, la chiamata a `credentialsProvider.identityId` restituirà `nil`. Per ulteriori informazioni, consulta la documentazione di riferimento del'SDK per iOS.

iOS - Swift

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui i passaggi seguenti.

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication in Amplify Dev Center.

Recupero di una identità di Amazon Cognito

Puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale se autorizzi gli utenti non autenticati o dopo aver impostato i token di accesso nel fornitore di credenziali se autentichi gli utenti:

```
// Retrieve your Amazon Cognito ID
credentialsProvider.getIdentityId().continueWith(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` è una chiamata asincrona. Se sul tuo provider è già impostato un ID identità, puoi chiamare `credentialsProvider.identityId` per recuperare tale identità, la quale è memorizzata nella cache in locale. Tuttavia, se sul provider non è impostato alcun ID identità, la chiamata a `credentialsProvider.identityId` restituirà `nil`. Per ulteriori informazioni, consulta la documentazione di riferimento del'SDK per iOS.

JavaScript

Se non lo hai già fatto, crea un pool di identità nella console Amazon Cognito prima di utilizzare `AWS.CognitoIdentityCredentials`.

Dopo aver configurato un pool di identità con i provider di identità, puoi utilizzare `AWS.CognitoIdentityCredentials` per autenticare gli utenti. Per configurare le credenziali dell'applicazione per utilizzare `AWS.CognitoIdentityCredentials`, imposta la proprietà `credentials` di `AWS.Config` o di una configurazione per servizio. Nell'esempio seguente viene utilizzato `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 proprietà `Logins` opzionale è una mappa di nomi di provider di identità ai token di identità per tali provider. Il modo in cui ottieni il token dal provider di identità dipende dal provider utilizzato. Ad esempio, se Facebook è uno dei provider di identità, puoi utilizzare la funzione `FB.login` di SDK di Facebook per ottenere un token del provider di identità:

```
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.');
    }
});
```

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale. Se autentichi gli utenti, lo puoi recuperare dopo aver impostato i token di accesso nel fornitore di credenziali:

```
var identityId = AWS.config.credentials.identityId;
```

Unità

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione, in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui i passaggi seguenti.

L'SDK AWS per Xamarin fa ora parte di AWS SDK for .NET. Per iniziare a usare Amazon Cognito in AWS SDK for .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK for .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale. Se autentichi gli utenti, lo puoi recuperare dopo aver impostato i token di accesso nel fornitore di credenziali:

```
credentials.GetIdentityIdAsync(delegate(AmazonCognitoIdentityResult<string> result) {
    if (result.Exception != null) {
        //Exception!
    }
    string identityId = result.Response;
});
```

Xamarin

Puoi usare Amazon Cognito per fornire credenziali temporanee con privilegi limitati alla tua applicazione in modo che gli utenti possano accedere alle risorse. AWS Amazon Cognito supporta sia le identità autenticate sia le identità non autenticate. Per fornire AWS le credenziali alla tua app, segui i passaggi seguenti.

L'SDK AWS per Xamarin fa ora parte di AWS SDK for .NET. Per iniziare a usare Amazon Cognito in AWS SDK for .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK for .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

> **Note**
>
> Nota: se hai creato il pool di identità prima di febbraio 2015, devi riassociare i ruoli con il pool di identità, in modo da utilizzare questo costruttore senza i ruoli come parametri. A tale scopo, apri console Amazon Cognito, scegli Manage identity pools (Gestisci pool di identità), seleziona il pool di identità, scegli Edit identity Pool (Modifica pool di identità), specifica i ruoli autenticati e non autenticati e salva le modifiche.

Recupero di una identità di Amazon Cognito

Se autorizzi gli utenti non autenticati, puoi recuperare immediatamente un identificatore di identità univoco di Amazon Cognito (ID identità) per l'utente finale. Se autentichi gli utenti, lo puoi recuperare dopo aver impostato i token di accesso nel fornitore di credenziali:

```
var identityId = await credentials.GetIdentityIdAsync();
```

Accesso ai servizi AWS

Dopo aver configurato il provider di credenziali Amazon Cognito e recuperato le AWS credenziali, puoi creare un client. Servizio AWS

AWS Risorse SDK per la creazione di un client

- AWS Configurazione del client nella Guida per gli AWS SDK for C++ sviluppatori
- Utilizzo della AWS SDK for Go versione 2 con la Servizi AWS Guida per gli AWS SDK for Go sviluppatori
- Configurazione dei client HTTP nella Developer Guide AWS SDK for Java 2.x
- Creazione e chiamata di oggetti di servizio nella AWS SDK for JavaScript Developer Guide
- Creazione di client nella AWS SDK for Python (Boto3) documentazione
- Creazione di un client di servizio nella Guida per AWS SDK for Rust gli sviluppatori
- Utilizzo dei client nella Guida per SDK AWS per Swift gli sviluppatori

Ad esempio, il seguente frammento inizializza un client Amazon DynamoDB:

Android

Per utilizzare un pool di identità di Amazon Cognito in un'app Android, configura. AWS Amplify Per ulteriori informazioni, consultare Autenticazione in Amplify Dev Center.

```
// Create a service client with the provider
AmazonDynamoDB client = new AmazonDynamoDBClient(credentialsProvider);
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

iOS - Objective-C

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication e Flutter Authentication nella 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];
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

iOS - Swift

Per utilizzare un pool di identità di Amazon Cognito in un'app iOS, configura. AWS Amplify Per ulteriori informazioni, consultare Swift Authentication in 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")
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

JavaScript

```
// Create a service client with the provider
var dynamodb = new AWS.DynamoDB({region: 'us-west-2'});
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

Unità

L'SDK AWS per Xamarin fa ora parte di AWS SDK for .NET. Per iniziare a usare Amazon Cognito in AWS SDK for .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK for .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
AmazonDynamoDBClient client = new AmazonDynamoDBClient(credentials, REGION);
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

Xamarin

L'SDK AWS per Xamarin fa ora parte di AWS SDK for .NET. Per iniziare a usare Amazon Cognito in AWS SDK for .NET, consulta il fornitore di credenziali Amazon Cognito nella Developer Guide. AWS SDK for .NET Oppure consulta Amplify Dev Center per le opzioni con cui creare un'app. AWS Amplify

```
// create a service client that uses credentials provided by Cognito
```

```
var client = new AmazonDynamoDBClient(credentials, REGION)
```

Il fornitore di credenziali comunica con Amazon Cognito, recuperando sia l'identificatore univoco per gli utenti autenticati e non autenticati sia le credenziali temporanee con privilegi limitati per Mobile SDK. AWS AWS Le credenziali recuperate sono valide per un'ora e quando scadono vengono aggiornate dal fornitore.

Provider di identità esterni con pool di identità

Con la proprietà `logins` puoi impostare le credenziali ricevute da un provider di identità. Inoltre, puoi associare un pool di identità a più IdPs di un pool. Ad esempio, puoi impostare i token di Facebook e Google nella proprietà `logins` per associare l'identità univoca di Amazon Cognito agli accessi di entrambi i provider di identità. L'utente può eseguire l'autenticazione con un dei due account, ma Amazon Cognito restituisce lo stesso identificatore utente.

Le seguenti istruzioni ti guidano nell'autenticazione con i pool di identità IdPs supportati dai pool di identità di Amazon Cognito.

Argomenti

- Configurazione di Facebook come pool di identità IdP
- Configurazione di Login with Amazon come IdP di pool di identità
- Configurazione di Google come IdP del pool di identità
- Configurazione di Accedi con Apple come IdP del pool di identità
- Configurazione di un provider OIDC come pool di identità IdP
- Configurazione di un provider SAML come IdP del pool di identità

Configurazione di Facebook come pool di identità IdP

I pool di identità di Amazon Cognito si integrano con Facebook per fornire l'autenticazione federata per gli utenti della tua Webe per dispositivi mobili. Questa sezione illustra come registrare e impostare la tua applicazione con Facebook come provider di identità.

Impostazione di Facebook

Prima di autenticare gli utenti di Facebook e interagire con le API di Facebook, è necessario registrare la tua applicazione su Facebook.

Il portale per gli sviluppatori di Facebook ti aiuta a configurare l'applicazione. Esegui questa procedura prima di integrare Facebook nel tuo pool di identità Amazon Cognito:

Configurazione di Facebook

1. Nel portale per sviluppatori di Facebook, esegui l'accesso con le credenziali di Facebook.
2. Nel menu Apps (App), scegli Add a New App (Aggiungi una nuova app).
3. Scegli una piattaforma e completa la procedura di avvio rapido.

Android

Per ulteriori informazioni su come integrare le app Android con Facebook Login, consulta la guida Primi passi di Facebook.

iOS - Objective-C

Per ulteriori informazioni su come integrare le app iOS Objective-C con Facebook Login, consulta la guida Primi passi di Facebook.

iOS - Swift

Per ulteriori informazioni su come integrare le app iOS Swift con Facebook Login, consulta la guida Primi passi di Facebook.

JavaScript

Per ulteriori informazioni su come integrare le app JavaScript Web con Facebook Login, consulta la Guida introduttiva di Facebook.

Unità

Per ulteriori informazioni su come integrare le app Unity con Facebook Login, consulta la guida Primi passi di Facebook.

Xamarin

Per aggiungere l'autenticazione di Facebook, segui innanzitutto il flusso appropriato di seguito per integrare l'SDK di Facebook nella tua applicazione. I pool di identità di Amazon Cognito usano il token di accesso di Facebook per generare un identificatore utente univoco associato a un'identità di Amazon Cognito.

- SDK di Facebook per iOS di Xamarin
- SDK di Facebook per Android di Xamarin

Configurazione di un provider di identità nella console dei pool di identità di Amazon Cognito

Per configurare il provider di identità, utilizza la procedura seguente.

Per aggiungere un gestore dell'identità digitale Facebook

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Facebook.
5. Inserisci l'ID app del progetto OAuth creato su Meta per sviluppatori. Per ulteriori informazioni, consultare Facebook Login nella documentazione Meta per sviluppatori.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
        i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
        ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.
    a. Per non applicare alcun tag principale, scegli Inattivo.
    b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Facebook

Android

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente di Android. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token di accesso di questo oggetto di sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di Amazon Cognito.

SDK di Facebook 4.0 o versione successiva:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", AccessToken.getCurrentAccessToken().getToken());
credentialsProvider.setLogins(logins);
```

Versione precedente a SDK di Facebook 4.0:

```
Map<String, String> logins = new HashMap<String, String>();
logins.put("graph.facebook.com", Session.getActiveSession().getAccessToken());
credentialsProvider.setLogins(logins);
```

Il processo di accesso di Facebook inizializza una sessione singleton nell'SDK. L'oggetto sessione di Facebook contiene un token OAuth che Amazon Cognito utilizza per AWS generare credenziali per l'utente finale autenticato. Amazon Cognito usa inoltre il token per controllare la presenza di un utente corrispondente a questa particolare identità di Facebook nel tuo database degli utenti. Se l'utente esiste già, l'API restituisce l'identificatore esistente. In caso contrario, l'API restituisce un nuovo identificatore. Gli identificatori vengono automaticamente memorizzati nella cache dall'SDK client sul dispositivo locale.

> **Note**
>
> Dopo aver impostato la mappa degli accessi, effettua una chiamata o recupera `refresh` le credenziali. `get` AWS

iOS - Objective-C

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito usa il token di accesso da questo oggetto di sessione per autenticare l'utente e associarlo a un pool di identità di Amazon Cognito univoco (identità federate).

Per fornire il token di accesso di Facebook ad Amazon Cognito, implementa il protocollo AWSIdentityProviderManager.

Quando implementi il metodo `logins`, viene restituito un dizionario contenente `AWSIdentityProviderFacebook`. Questo dizionario funge da chiave e il token di accesso corrente dell'utente Facebook autenticato funge da valore, come mostrato nell'esempio di codice seguente.

```
- (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"}]];
    }
}
```

Quando crei istanze di `AWSCognitoCredentialsProvider`, trasmetti la classe che implementa `AWSIdentityProviderManager` come il valore di `identityProviderManager` nel costruttore. Per ulteriori informazioni, vai alla pagina di AWSCognitoCredentialsProviderriferimento e scegli initWithRegionTipo::. identityPoolId identityProviderManager

iOS - Swift

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Quindi aggiungi un pulsante Accedi con Facebook alla tua interfaccia utente. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito usa il token di accesso da questo oggetto di sessione per autenticare l'utente e associarlo a un pool di identità di Amazon Cognito univoco (identità federate).

Per fornire il token di accesso di Facebook ad Amazon Cognito, implementa il protocollo AWSIdentityProviderManager.

Quando implementi il metodo `logins`, viene restituito un dizionario contenente `AWSIdentityProviderFacebook`. Questo dizionario funge da chiave e il token di accesso corrente dell'utente Facebook autenticato funge da valore, come mostrato nell'esempio di codice seguente.

```
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"]))
    }
}
```

Quando crei istanze di `AWSCognitoCredentialsProvider`, trasmetti la classe che implementa `AWSIdentityProviderManager` come il valore di `identityProviderManager` nel costruttore. Per ulteriori informazioni, vai alla pagina AWSCognitoCredentialsProviderdi riferimento e scegli initWithRegionTipo:identityPoolId: identityProviderManager.

JavaScript

Per fornire l'autenticazione di Facebook, segui le istruzioni in Facebook Login per il web per aggiungere il pulsante Accedi con Facebook sul tuo sito Web. L'SDK di Facebook utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token di accesso di questo oggetto di sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di 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.');
  }

});
```

L'SDK di Facebook ottiene un token OAuth che Amazon Cognito utilizza per generare AWS credenziali per l'utente finale autenticato. Amazon Cognito usa inoltre il token per controllare la presenza di un utente corrispondente a questa particolare identità di Facebook nel tuo database utenti. Se l'utente esiste già, l'API restituisce l'identificatore esistente. In caso contrario, viene restituito un nuovo identificatore. Gli identificatori vengono automaticamente memorizzati nella cache dall'SDK client sul dispositivo locale.

> **Note**
>
> Dopo aver impostato la mappa degli accessi, è necessario effettuare una chiamata a `refresh` o `get` per ottenere le credenziali. Per un esempio di codice, vedi «Caso d'uso 17, Integrazione di pool di utenti con Cognito Identity» nel JavaScript file README.

Unità

Per aggiungere l'autenticazione di Facebook, segui innanzitutto la guida di Facebook per integrare l'SDK di Facebook nella tua applicazione. Amazon Cognito usa il token di accesso di Facebook dall'oggetto FB per generare un identificatore utente univoco associato a un'identità di Amazon Cognito.

Una volta autenticato l'utente con l'SDK di Facebook, aggiungi il token di sessione al provider di credenziali di 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);
}
```

Prima di utilizzare `FB.AccessToken`, chiama `FB.Login()` e assicurati che `FB.IsLoggedIn` sia true.

Xamarin

Xamarin per Android:

```
public void InitializeFacebook() {
    FacebookSdk.SdkInitialize(this.ApplicationContext);
    callbackManager = CallbackManagerFactory.Create();
    LoginManager.Instance.RegisterCallback(callbackManager, new FacebookCallback &lt;
 LoginResult &gt; () {
      HandleSuccess = loginResult = &gt; {
        var accessToken = loginResult.AccessToken;
        credentials.AddLogin("graph.facebook.com", accessToken.Token);
        //open new activity
      },
      HandleCancel = () = &gt; {
        //throw error message
      },
      HandleError = loginError = &gt; {
        //throw error message
      }
    });
    LoginManager.Instance.LogInWithReadPermissions(this, new List &lt; string &gt; {
      "public_profile"
    });
  }
```

Xamarin per 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
    }
  });
}
```

Configurazione di Login with Amazon come IdP di pool di identità

Amazon Cognito si integra con Login with Amazon per fornire l'autenticazione federata per gli utenti della tua app Web e per dispositivi mobili. Questa sezione spiega come registrare e impostare la tua applicazione con Login with Amazon come provider di identità.

Configura Login with Amazon per l'uso di Amazon Cognito nel portale per sviluppatori. Per ulteriori informazioni, consulta Configurazione di Login with Amazon nelle domande frequenti (FAQ) di Login with Amazon.

> **Note**
>
> Per integrare Login with Amazon in un'applicazione Xamarin, segui la Guida introduttiva di Xamarin.

> **Note**
>
> Non puoi integrare in modo nativo Login with Amazon nella piattaforma Unity. Puoi utilizzare invece una visualizzazione Web e seguire il flusso di accesso del browser.

Configurazione di Login with Amazon

Implementazione di Login with Amazon

Nel portale per gli sviluppatori Amazon, puoi configurare un'applicazione OAuth da integrare con il tuo pool di identità, trovare la documentazione Login with Amazon e scaricare gli SDK. Scegli Developer console (Console per sviluppatori), quindi Login with Amazon nel portale per gli sviluppatori. Puoi creare un profilo di sicurezza per la tua applicazione e quindi costruire meccanismi di autenticazione Login with Amazon nella tua app. Consulta Ottenere le credenziali per ulteriori informazioni su come integrare l'autenticazione Login with Amazon con la tua app.

Amazon genera un ID client OAuth 2.0 per il tuo nuovo profilo di sicurezza. Puoi trovare l'ID client nella scheda Web Settings (Impostazioni Web) del profilo di sicurezza. Inserisci l'ID del profilo di sicurezza nel campo ID app dell'IdP Login with Amazon nel pool di identità.

> **Note**
>
> Inserisci l'ID del profilo di sicurezza nel campo ID app dell'IdP Login with Amazon nel pool di identità. Ciò differisce dai pool di utenti, che utilizzano l'ID client.

Configurazione del provider esterno nella console di Amazon Cognito

Per aggiungere un gestore dell'identità digitale Login with Amazon

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli Login with Amazon.
5. Inserisci l'ID app del progetto OAuth creato su Login with Amazon. Per ulteriori informazioni, consultare la documentazione di Login with Amazon.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.
   a. Per non applicare alcun tag principale, scegli Inattivo.
   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Login with Amazon: Android

Dopo aver autenticato Amazon Login, puoi passare il token al provider di credenziali Amazon Cognito nel metodo OnSuccess dell'interfaccia. TokenListener Il codice sarà il seguente:

```
@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);
}
```

Utilizzo di Login with Amazon: iOS - Objective-C

Dopo aver autenticato l'accesso ad Amazon, puoi passare il token al provider di credenziali Amazon Cognito con requestDidSucceed il metodo 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 };
    }
}}
```

Utilizzo di Login with Amazon: iOS - Swift

Una volta autenticato l'accesso ad Amazon, puoi passare il token al provider di credenziali di Amazon Cognito nel metodo `requestDidSucceed` di `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]
    }
}
```

Usa Login with Amazon: JavaScript

Dopo che l'utente ha effettuato l'autenticazione con Login with Amazon e viene reindirizzato al tuo sito Web, il token di accesso di Login with Amazon viene fornito nella stringa di query. Passa tale token alla mappa degli accessi delle credenziali.

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
   IdentityPoolId: 'IDENTITY_POOL_ID',
   Logins: {
       'www.amazon.com': 'Amazon Access Token'
   }
});
```

Utilizzo di Login with Amazon: Xamarin

Xamarin per 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 per iOS

In `AppDelegate.cs`, inserire quanto segue:

```
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;
}
```

Quindi in `ViewController.cs`, procedere come segue:

```
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 : AIAuthenticationDelegate
{
  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());
    }
}
```

Configurazione di Google come IdP del pool di identità

Amazon Cognito si integra con Google per fornire l'autenticazione federata per gli utenti della tua applicazione per dispositivi mobili. Questa sezione spiega come registrare e impostare la tua applicazione con Google come provider di identità.

Android

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurarla come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità di più client di Google, consulta l'articolo Cross-client Identity.

Configurazione di Google

Per attivare Google Sign-in per Android, è necessario creare un progetto di console di Google Developers per l'applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona Android per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio, quindi scegli Create and continue (Crea e continua).

5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app Android, consulta Autenticare gli utenti con Accedi con Google nella documentazione di Google Identity.

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
        i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
        ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

a. Per non applicare alcun tag principale, scegli Inattivo.

b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui le istruzioni della documentazione di Google per Android. Quando un utente esegue l'accesso, viene richiesto un token di autenticazione OpenID Connect a Google. Quindi, Amazon Cognito utilizza il token per autenticare l'utente e generare un identificatore univoco.

Il seguente codice di esempio illustra come recuperare il token di autenticazione da 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**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità di più client di Google, consulta l'articolo Cross-client Identity.

Configurazione di Google

Per abilitare Google Sign-in per iOS, è necessario creare un progetto di console di Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona iOS per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio. Scegli la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni sull'integrazione di Google nell'app iOS, consulta Google Sign-In per iOS nella documentazione di Google Identity.

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.

6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per iOS. In caso di autenticazione corretta, si genera un token di autenticazione di OpenID Connect, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco.

In caso di autenticazione corretta, viene generato un oggetto `GTMOAuth2Authentication` contenente un `id_token`, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco:

```
- (void)finishedWithAuth: (GTMOAuth2Authentication *)auth error: (NSError *) error {
        NSString *idToken = [auth.parameters objectForKey:@"id_token"];
```

```
        credentialsProvider.logins = @{ @(AWSCognitoLoginProviderKeyGoogle): idToken };
    }
```

iOS - Swift

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità di più client di Google, consulta l'articolo Cross-client Identity.

Configurazione di Google

Per abilitare Google Sign-in per iOS, è necessario creare un progetto di console di Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona iOS per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni sull'integrazione di Google nell'app iOS, consulta Google Sign-In per iOS nella documentazione di Google Identity.

Scegli Gestisci pool di identità dalla home page della console Amazon Cognito:

Configurazione del provider esterno nella console Amazon Cognito

1. Scegli il nome del pool di identità per cui vuoi abilitare Google come provider esterno. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), scegli Edit identity pool (Modifica pool di identità). Viene visualizzata la pagina Edit identity pool (Modifica pool di identità).
3. Scorri verso il basso e scegli Authentication providers (Provider di autenticazione) per espandere la sezione.
4. Scegli la scheda Google .
5. Scegli Unlock (Sblocca).
6. Inserisci l'ID client ottenuto da Google, quindi scegli Save Changes (Salva le modifiche).

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per iOS. In caso di autenticazione corretta, viene generato un token di autenticazione OpenID Connect, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco.

L'autenticazione completata restituisce un oggetto `GTMOAuth2Authentication` contenente un `id_token`. Amazon Cognito utilizza questo token per autenticare l'utente e generare un identificatore univoco:

```
func finishedWithAuth(auth: GTMOAuth2Authentication!, error: NSError!) {
    if error != nil {
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.parameters.objectForKey("id_token")
      credentialsProvider.logins = [AWSCognitoLoginProviderKey.Google.rawValue:
 idToken!]
    }
}
```

JavaScript

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità di più client di Google, consulta l'articolo Cross-client Identity.

Configurazione di Google

Per abilitare Google Sign-in per un'app JavaScript web, crea un progetto di console Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando chiede il consenso a condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona Web application (Applicazione Web) per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
5. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
6. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Per ulteriori informazioni su come integrare Google nella tua app Web, consulta Sign In With Google nella documentazione di Google Identity.

Configurazione del provider esterno nella console Amazon Cognito

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.
   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.
     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.
     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.
7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.
   a. Per non applicare alcun tag principale, scegli Inattivo.
   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.
   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.
8. Seleziona Salva modifiche.

Utilizzo di Google

Per abilitare l'accesso con Google nella tua applicazione, segui la documentazione Google per Web.

In caso di autenticazione corretta, viene generato un oggetto di risposta contenente un `id_token`, utilizzato da Amazon Cognito per autenticare l'utente e generare un identificatore univoco:

```
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.
     });
  }
}
```

Unità

Configurazione di Google

Per abilitare Google Sign-in per un'app Unity, è necessario creare un progetto di console di Google Developers per la tua applicazione.

1. Vai alla console di Google Developers e crea un nuovo progetto.
2. Scegli APIs & Services (API e servizi), quindi OAuth consent screen (Schermata del consenso OAuth). Personalizza le informazioni che Google mostra agli utenti quando richiede il consenso per condividere i dati del profilo con la tua app.
3. Scegli Credentials (Credenziali), quindi Create credentials (Crea credenziali). Scegli OAuth client ID (ID client OAuth). Seleziona Web application (Applicazione Web) per Application type (Tipo di applicazione). Crea un ID client separato per ogni piattaforma in cui sviluppi la tua app.
4. Per Unity, crea un ulteriore ID client OAuth per Android e un altro per iOS.

5. Da Credentials (Credenziali), scegli Manage service accounts (Gestione degli account del servizio). Scegli Create service account (Crea account del servizio). Inserisci i dettagli dell'account del servizio e scegli Create and continue (Crea e continua).
6. Concedi all'account del servizio l'accesso al tuo progetto. Concedi agli utenti l'accesso all'account del servizio come richiesto dall'app.
7. Scegli il tuo nuovo account del servizio, seleziona la scheda Keys (Chiavi) e Add key (Aggiungi chiave). Crea e scarica una nuova chiave JSON.

Per ulteriori informazioni su come utilizzare la console di Google Developers, consulta Creating and managing projects (Creazione e gestione di progetti) nella documentazione di Google Cloud.

Creazione di un provider OpenID nella console IAM

1. Crea un provider OpenID nella console IAM. Per istruzioni su come configurare un provider OpenID, consulta Utilizzo dei provider di identità OpenID Connect.
2. Quando viene richiesto l'URL del tuo provider, immetti `"https://accounts.google.com"`.
3. Quando viene richiesto di immettere un valore nel campo Destinatari, immetti uno dei tre ID client creati nelle fasi precedenti.
4. Scegli il nome del provider e aggiungi due ulteriori destinatari con gli altri due ID client.

Configurazione del provider esterno nella console Amazon Cognito

Scegli Gestisci pool di identità dalla home page della console Amazon Cognito:

Per aggiungere un gestore dell'identità digitale di Google

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Seleziona Google.
5. Inserisci l'ID cliente del progetto OAuth creato su Google Cloud Platform. Per ulteriori informazioni, consultare Configurazione di OAuth 2.0 della Guida in linea della console di Google Cloud Platform.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

- Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

  i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

  ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Installazione del plugin di Google per Unity

1. Aggiungi il plugin Google Play Giochi per Unity al tuo progetto Unity.
2. In Unity, dal menu di Windows configura il plug-in utilizzando i tre ID per le piattaforme Android e iOS.

Utilizzo di Google

Il seguente codice di esempio illustra come recuperare il token di autenticazione da 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 non supporta in modo nativo Google sulla piattaforma Xamarin. Richiede attualmente l'uso di una visualizzazione Web per seguire il flusso di accesso del browser. Per ulteriori informazioni su come l'integrazione di Google funziona con altri SDK, scegli un'altra piattaforma.

Per abilitare l'accesso con Google nella tua applicazione, autentica gli utenti e recupera un token OpenID Connect. Amazon Cognito usa questo token per generare un identificatore utente univoco associato a un'identità di Amazon Cognito. Purtroppo, l'SDK di Google per Xamarin non consente di recuperare il token OpenID Connect, perciò è necessario utilizzare un client alternativo o il flusso Web in una visualizzazione Web.

Dopo aver ottenuto il token, puoi impostarlo in `CognitoAWSCredentials`:

```
credentials.AddLogin("accounts.google.com", token);
```

> **Note**
>
> Se l'app utilizza Google ed è disponibile su più piattaforme per dispositivi mobili, è necessario configurare Google come provider OpenID Connect. Aggiungi tutti gli ID client creati come valori di destinatari aggiuntivi per consentire una migliore integrazione. Per ulteriori informazioni sul modello di identità di più client di Google, consulta l'articolo Cross-client Identity.

Configurazione di Accedi con Apple come IdP del pool di identità

Amazon Cognito si integra con Accesso con Apple per fornire l'autenticazione federata per gli utenti della tua applicazione Web e per dispositivi mobili. Questa sezione spiega come registrare e configurare l'applicazione con Accedi con Apple come provider di identità.

Per aggiungere Accedi con Apple come provider di autenticazione a un pool di identità, devi completare due procedure. Innanzitutto devi integrare Accedi con Apple in un'applicazione, quindi configurare Accedi con Apple nei pool di identità. Per la maggior parte delle up-to-date informazioni sulla configurazione di Sign in with Apple, consulta Configurazione dell'ambiente per l'accesso con Apple nella documentazione per gli sviluppatori Apple.

Configurazione di Accedi con Apple

Per configurare Accedi con Apple come provider di identità, è necessario registrare l'applicazione con Apple per ricevere l'ID client.

1. Creazione di un account sviluppatore con Apple.
2. Accedi con le tue credenziali Apple.
3. Nel riquadro di navigazione a sinistra, scegliere Certificates, IDs & Profiles (Certificati, identificatori e profili).
4. Nel riquadro di navigazione a sinistra, scegliere Inputs (Input).
5. Nella pagina Identificatori scegli l'icona +.
6. Nella pagina Register a New Identifier (Registra un nuovo identificatore), scegli App IDs (ID app), quindi scegli Continue (Continua).
7. Nella pagina Register an App ID (Registra un'ID app), procedi come indicato di seguito:

    a. In Description (Descrizione), digitare una descrizione.

b. In Bundle ID (ID pacchetto) digitare un identificatore. Prendi nota dell'ID bundle poiché avrai bisogno di questo valore per configurare Apple come provider nel pool di identità.

c. In Capabilities (Funzionalità), scegli Sign In with Apple (Accedi con Apple), quindi seleziona Edit (Modifica).

d. Nella pagina Accedi con Apple: configurazione dell'ID Apple, seleziona l'impostazione appropriata per l'app. Quindi scegli Save (Salva).

e. Scegli Continua.

8. Nella pagina Confirm your App ID (Conferma l'ID app), scegli Register (Registra).
9. Passa alla fase 10 se desideri integrare Accedi con Apple con un'applicazione iOS nativa. La fase 11 è per le applicazioni che si desidera integrare con Accedi con Apple JS.
10. Nella pagina Identifiers (Identificatori), scegli il menu App IDs (ID app), quindi Services IDs (ID servizio). Scegli l'icona +.
11. Nella pagina Register a New Identifier (Registra un nuovo identificatore), seleziona Services IDs (ID servizi), quindi scegli Continue (Continua).
12. Nella pagina Register a Services ID (Registra un ID servizi), procedi nel modo seguente:

    a. In Description (Descrizione), digitare una descrizione.

    b. In Identifier (Identificatore), digitare un identificatore. Prendi nota dell'ID servizio poiché avrai bisogno di questo valore per configurare Apple come provider nel pool di identità.

    c. Scegli Accedi con Apple, quindi scegliere Configure (Configura).

    d. Nella pagina Web Authentication Configuration (Configurazione autenticazione Web), scegli un Primary App ID (ID app principale). In Website URLs (URL del sito Web), scegli l'icona +. Per Domini e sottodomini, inserire il nome di dominio dell'app. In Return URLs (URL restituiti) inserisci l'URL di callback a cui l'autorizzazione reindirizza l'utente dopo l'autenticazione con Accedi con Apple.

    e. Seleziona Successivo.

    f. Scegli Continue (Continua), quindi scegli Register (Registra).

13. Nel riquadro di navigazione a sinistra, scegliere Chiavi.
14. Nella pagina Keys (Chiavi), scegli l'icona +.
15. Nella pagina Register a New Key, (Registra una nuova chiave) procedi nel modo seguente:

    a. In Key Name (Nome chiave), digitare un nome della chiave.

    b. Seleziona Accedi con Apple, quindi seleziona Configura.

c. Nella pagina Configura chiave, scegliere un ID app principale, quindi scegli Salva.

d. Scegli Continue (Continua), quindi scegli Register (Registra).

> **Note**
>
> Per integrare Accedi con Apple con un'applicazione iOS nativa, consulta Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple).
> Per integrare Accedi con Apple in una piattaforma diversa da iOS nativo, consulta Sign in with Apple JS (Accedi con Apple JS).

Configurazione del provider esterno nella console delle identità federate di Amazon Cognito

Per configurare il provider esterno, utilizzare la procedura seguente.

Per aggiungere un gestore dell'identità digitale Accedi con Apple

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli Accedi con Apple.
5. Inserisci l'ID servizi del progetto OAuth creato con Apple Developer. Per ulteriori informazioni, consultare la pagina relativa all'autenticazione degli utenti con Accedi con Apple nella documentazione di Accedi con Apple.
6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando

l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Accesso con Apple come provider negli esempi della CLI di identità federate di Amazon Cognito

In questo esempio viene creato un pool di identità denominato `MyIdentityPool` con Accedi con Apple come provider di identità.

```
aws cognito-identity create-identity-pool --identity-
pool-name MyIdentityPool --supported-login-providers
appleid.apple.com="sameple.apple.clientid"
```

Per ulteriori informazioni, consulta l'argomento relativo alla creazione di un pool di identità

Genera un ID identità di Amazon Cognito

Questo esempio genera (o recupera) un ID Amazon Cognito. Questa è un'API pubblica, quindi non sono necessarie credenziali per richiamare questa API.

```
aws cognito-identity get-id --identity-pool-id SampleIdentityPoolId --
logins appleid.apple.com="SignInWithAppleIdToken"
```

Per ulteriori informazioni, consulta get-id.

Ottenimento delle credenziali per un ID identità di Amazon Cognito

In questo esempio vengono restituite le credenziali per l'ID di identità fornito e per l'accesso con Accedi con Apple. Questa è un'API pubblica, quindi non sono necessarie credenziali per richiamare questa API.

```
aws cognito-identity get-credentials-for-identity --identity-id
SampleIdentityId --logins appleid.apple.com="SignInWithAppleIdToken"
```

Per ulteriori informazioni, vedi get-credentials-for-identity

Utilizzo di Accedi con Apple: Android

Apple non fornisce un SDK che supporta Accedi con Apple per Android. È invece possibile utilizzare il flusso Web in una visualizzazione Web.

- Per configurare Accedi con Apple nell'applicazione, seguire le indicazioni riportate in Configuring Your Webpage for Sign In with Apple (Configurazione della pagina Web per Accedi con Apple) nella documentazione Apple.
- Per aggiungere un pulsante Accedi con Apple all'interfaccia utente Android, segui le indicazioni riportate in Displaying and Configuring Sign In with Apple Buttons (Visualizzazione e configurazione dell'accesso con i pulsanti Apple) nella documentazione Apple.
- Per autenticare in modo sicuro gli utenti utilizzando il servizio Accedi con Apple, segui le indicazioni in Autenticazione degli utenti con Accedi con Apple nella documentazione Apple (lingua italiana non garantita).

L'accesso con Apple utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token ID di questo oggetto sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. 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);
}
```

Utilizzo di Accedi con Apple: iOS - Objective-C

Apple ha fornito il supporto SDK per Accedi con Apple nelle applicazioni iOS native. Per implementare l'autenticazione utente con Accedi con Apple nei dispositivi iOS nativi, seguire le indicazioni riportate in Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple) nella documentazione di Apple.

Amazon Cognito utilizza il token ID per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

```
(void)finishedWithAuth: (ASAuthorizationAppleIDCredential *)auth error: (NSError *)
 error {
        NSString *idToken = [ASAuthorizationAppleIDCredential
 objectForKey:@"identityToken"];
        credentialsProvider.logins = @{ "appleid.apple.com": idToken };
    }
```

Utilizzo di Accedi con Apple: iOS - Swift

Apple ha fornito il supporto SDK per Accedi con Apple nelle applicazioni iOS native. Per implementare l'autenticazione utente con Accedi con Apple nei dispositivi iOS nativi, seguire le indicazioni riportate in Implementing User Authentication with Sign in with Apple (Implementazione dell'autenticazione utente con Accedi con Apple) nella documentazione di Apple.

Amazon Cognito utilizza il token ID per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Per ulteriori informazioni sulla configurazione di Accedi con Apple in iOS, consulta Configurazione di Accedi con Apple (lingua italiana non garantita).

```
func finishedWithAuth(auth: ASAuthorizationAppleIDCredential!, error: NSError!) {
    if error != nil {
      print(error.localizedDescription)
    }
    else {
      let idToken = auth.identityToken,
      credentialsProvider.logins = ["appleid.apple.com": idToken!]
    }
}
```

Usa Accedi con Apple: JavaScript

Apple non fornisce un SDK che supporti l'accesso con Apple per JavaScript. È invece possibile utilizzare il flusso Web in una visualizzazione Web.

- Per configurare Accedi con Apple nell'applicazione, seguire le indicazioni riportate in Configuring Your Webpage for Sign In with Apple (Configurazione della pagina Web per Accedi con Apple) nella documentazione Apple.
- Per aggiungere un pulsante Accedi con Apple alla tua interfaccia JavaScript utente, segui Visualizzazione e configurazione dell'accesso con i pulsanti Apple nella documentazione Apple.
- Per autenticare in modo sicuro gli utenti utilizzando Accedi con Apple, seguire le indicazioni in Configurazione della pagina Web per Accedi con Apple nella documentazione Apple (lingua italiana non garantita).

L'accesso con Apple utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token ID di questo oggetto sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. 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.
     });
}
```

Utilizzo di Accedi con Apple: Xamarin

Non disponiamo di un SDK che supporta Accedi con Apple per Xamarin. È invece possibile utilizzare il flusso Web in una visualizzazione Web.

- Per configurare Accedi con Apple nell'applicazione, seguire le indicazioni riportate in Configurazione della pagina Web per Accedi con Apple nella documentazione Apple.
- Per aggiungere un pulsante Accedi con Apple all'interfaccia utente Xamarin, segui le indicazioni riportate in Displaying and Configuring Sign In with Apple Buttons (Visualizzazione e configurazione dell'accesso con i pulsanti Apple) nella documentazione Apple.
- Per autenticare in modo sicuro gli utenti utilizzando Accedi con Apple, seguire le indicazioni in Configurazione della pagina Web per Accedi con Apple nella documentazione Apple (lingua italiana non garantita).

L'accesso con Apple utilizza un oggetto di sessione per monitorare il suo stato. Amazon Cognito utilizza il token ID di questo oggetto sessione per autenticare l'utente, generare l'identificatore univoco e, se necessario, concedere all'utente l'accesso ad altre risorse. AWS

Dopo aver ottenuto il token, puoi impostarlo in `CognitoAWSCredentials`:

```
credentials.AddLogin("appleid.apple.com", token);
```

Configurazione di un provider OIDC come pool di identità IdP

OpenID Connect è uno standard aperto di autenticazione supportato da una serie di provider di accesso. Amazon Cognito supporta il collegamento delle identità con i provider OpenID Connect configurati tramite AWS Identity and Access Management.

Aggiunta di un provider OpenID Connect

Per informazioni su come creare un provider OpenID Connect, consultare Creazione di provider di identità OpenID Connect (OIDC) nella Guida per l'utente di AWS Identity and Access Management .

Associazione di un fornitore ad Amazon Cognito

Per aggiungere un gestore dell'identità digitale OIDC

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli OpenID Connect (OIDC).
5. Scegli un provider di identità OIDC dall'IAM del tuo. IdPs Account AWS Se desideri aggiungere un nuovo provider SAML, scegli Crea nuovo provider per accedere alla console IAM.

6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

    - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

        i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

        ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

    a. Per non applicare alcun tag principale, scegli Inattivo.

    b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

    c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

È possibile associare più provider OpenID Connect a un singolo pool di identità.

Utilizzo di OpenID Connect

Per informazioni su come accedere e ricevere un token ID, consulta la documentazione del provider.

Dopo aver ottenuto un token, aggiungilo alla mappa degli accessi. Usa l'URI del tuo provider come chiave.

Convalida di un token di OpenID Connect

Alla prima integrazione con Amazon Cognito, potresti ricevere un'eccezione `InvalidToken`. È importante comprendere in che modo Amazon Cognito convalida i token OpenID Connect (OIDC).

> **Note**
>
> Come specificato qui (https://tools.ietf.org/html/rfc7523), Amazon Cognito fornisce un periodo di tolleranza di 5 minuti per gestire qualsiasi sfasamento di ora tra sistemi.

1. Il parametro `iss` deve corrispondere alla chiave usata nella mappa degli accessi (ad esempio login.provider.com).
2. La firma deve essere valida. La firma deve essere verificabile mediante una chiave pubblica RSA.
3. L'impronta digitale della chiave pubblica del certificato corrisponde all'impronta digitale impostata in IAM al momento della creazione del provider OIDC.
4. In presenza del parametro `azp`, verifica questo valore rispetto agli ID client elencati nel provider OIDC.
5. In assenza del parametro `azp`, verifica il parametro `aud` rispetto agli ID client elencati nel provider OIDC.

Il sito Web jwt.io è una risorsa preziosa per la decodifica dei token e la verifica di questi valori.

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

Per fornire il token ID OIDC ad Amazon Cognito, implementa il protocollo `AWSIdentityProviderManager`.

Quando implementi il metodo `logins`, viene restituito un dizionario contenente il nome del provider OIDC configurato. Questo dizionario funge da chiave e il token ID corrente dell'utente autenticato funge da valore, come mostrato nell'esempio di codice seguente.

```
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")
    }
}
```

Quando crei un'istanza di`AWSCognitoCredentialsProvider`, passa la classe che implementa AWSIdentityProviderManager come valore di nel costruttore. identityProviderManager Per ulteriori informazioni, vai alla pagina di AWSCognitoCredentialsProviderriferimento e scegli initWithRegion Tipo::. identityPoolId identityProviderManager

JavaScript

```
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
 IdentityPoolId: 'IDENTITY_POOL_ID',
 Logins: {
    'login.provider.com': token
 }
});
```

Unità

```
credentials.AddLogin("login.provider.com", token);
```

Xamarin

```
credentials.AddLogin("login.provider.com", token);
```

Configurazione di un provider SAML come IdP del pool di identità

Amazon Cognito supporta l'autenticazione con provider di identità (IdPs) tramite Security Assertion Markup Language 2.0 (SAML 2.0). Puoi utilizzare un provider di identità che supporti il linguaggio SAML con Amazon Cognito per garantire un flusso di onboarding semplice per i tuoi utenti. Il tuo provider di identità supportato da SAML specifica i ruoli IAM che gli utenti possono assumere. In questo modo, diversi utenti possono ricevere diversi set di autorizzazioni.

Configurazione del pool di identità per un provider di identità SAML

La procedura seguente descrive come configurare il pool di identità per l'utilizzo di un provider di identità basato su SAML.

> **Note**
>
> Prima di configurare il pool di identità per supportare un provider SAML, è necessario configurare il provider di identità SAML nella console IAM. Per ulteriori informazioni, consulta Integrazione di provider di soluzioni SAML di terze parti con AWS nella Guida per l'utente di IAM.

Per aggiungere un gestore dell'identità digitale SAML

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli SAML.
5. Scegli un provider di identità SAML dall'IAM del tuo. IdPs Account AWS Se desideri aggiungere un nuovo provider SAML, scegli Crea nuovo provider per accedere alla console IAM.

6. Per impostare il ruolo richiesto da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Impostazioni ruolo.

   - Puoi assegnare agli utenti di tale IdP il ruolo predefinito impostato quando hai configurato il ruolo autenticato oppure selezionare l'opzione Scegli ruolo con regole.

     i. Se scegli l'opzione Scegli ruolo con regole, inserisci la Richiesta dall'autenticazione dell'utente, l'Operatore con cui desideri confrontare la richiesta, il Valore che determina una corrispondenza a questa scelta di ruolo e il Ruolo che desideri assegnare quando l'Assegnazione del ruolo corrisponde. Seleziona Aggiungi un altro per creare una regola aggiuntiva basata su una condizione diversa.

     ii. Scegli una Risoluzione del ruolo. Quando le richieste dell'utente non corrispondono alle regole, puoi negare le credenziali o emettere credenziali per il Ruolo autenticato.

7. Per modificare i tag principali assegnati da Amazon Cognito quando emette credenziali per gli utenti che hanno eseguito l'autenticazione con questo provider, configura Attributi per il controllo degli accessi.

   a. Per non applicare alcun tag principale, scegli Inattivo.

   b. Per applicare i tag principali in base alle richieste `sub` e `aud`, scegli Utilizza mappature predefinite.

   c. Per creare un tuo schema personalizzato di attributi dei tag principali, scegli Utilizza mappature personalizzate. Quindi, inserisci una Chiave tag che deve essere originata da ciascuna Richiesta che desideri rappresentare in un tag.

8. Seleziona Salva modifiche.

Configurazione di un provider di identità SAML

Una volta creato, configura il provider di identità SAML in modo da aggiungere una relazione di trust tra il provider di identità e AWS. Con molti IdPs, puoi specificare un URL che l'IdP può utilizzare per leggere le informazioni e i certificati del relying party da un documento XML. Per AWS, puoi usare https://signin.aws.amazon.com/static/saml-metadata.xml. Il passaggio successivo consiste nel configurare la risposta all'asserzione SAML del tuo IdP per compilare le attestazioni necessarie. AWS Per maggiori dettagli sulla configurazione delle richieste, consulta l'articolo sulla configurazione di asserzioni SAML per la risposta di autenticazione.

Quando l'IdP SAML include più di un certificato di firma nei metadati SAML, al momento dell'accesso il pool di utenti determina che l'asserzione SAML è valida se corrisponde a qualsiasi certificato nei metadati SAML.

Personalizzazione del ruolo utente con SAML

L'utilizzo di SAML con Amazon Cognito Identity ti consente la personalizzazione del ruolo per l'utente finale. Amazon Cognito supporta solo il flusso avanzato con il provider di identità basato su SAML. Non è necessario specificare un ruolo autenticato o non autenticato per consentire al pool di identità di utilizzare un provider di identità basato su SAML. L'attributo `https://aws.amazon.com/SAML/Attributes/Role` di registrazione specifica una o più coppie composte da un ARN di provider e da un ruolo delimitato da virgole. Questi sono i ruoli che l'utente può assumere. Puoi configurare il provider di identità SAML per popolare gli attributi di ruolo in base alle informazioni di attributo utente disponibili dal provider di identità. Se nell'asserzione SAML vengono ricevuti più ruoli, il parametro `customRoleArn` opzionale deve essere popolato quando chiami `getCredentialsForIdentity`. L'utente assume questo `customRoleArn` se il ruolo corrisponde a uno dell'attestazione nell'asserzione SAML.

Autenticazione di utenti con un provider di identità SAML

Per eseguire la federazione con l'IdP basato su SAML, determina l'URL a cui l'utente avvia l'accesso. AWS la federazione utilizza l'accesso avviato da IDP. In AD FS 2.0, l'URL prende la forma di `https://<fqdn>/adfs/ls/IdpInitiatedSignOn.aspx?loginToRp=urn:amazon:webservices`.

Per aggiungere il supporto per il provider di identità SAML in Amazon Cognito, è necessario prima autenticare gli utenti con il provider di identità SAML dalla tua applicazione iOS o Android. Il codice utilizzato per integrare e autenticare con il provider di identità SAML è specifico dei provider SAML. Dopo l'autenticazione dell'utente, puoi fornire l'asserzione SAML risultante ad Amazon Cognito Identity usando le API di Amazon Cognito.

Non puoi ripetere o riprodurre un'asserzione SAML nella mappa `Logins` della tua richiesta API del pool di identità. Un'asserzione SAML riprodotta dispone di un ID asserzione che duplica l'ID di una richiesta API precedente. Le operazioni API che possono accettare un'asserzione SAML nella `Logins` mappa includono GetId,, GetCredentialsForIdentitye ID. GetOpenIdTokenGetOpenTokenForDeveloperIdentity Puoi riprodurre un ID asserzione SAML una volta per richiesta API in un flusso di autenticazione del pool di identità. Ad esempio, puoi fornire la stessa asserzione SAML in una richiesta `GetId` e in una richiesta `GetCredentialsForIdentity` successiva, ma non in una seconda richiesta `GetId`.

Android

Se utilizzi l'SDK di Android, puoi popolare la mappa degli accessi con l'asserzione SAML come indicato di seguito.

```
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

Se stai utilizzando l'SDK di iOS, puoi fornire l'asserzione SAML in `AWSIdentityProviderManager` come indicato di seguito.

```
- (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";
}
```

Identità autenticate dagli sviluppatori (pool di identità)

Amazon Cognito supporta identità autenticate dagli sviluppatori oltre alla federazione delle identità web tramite Configurazione di Facebook come pool di identità IdP, Configurazione di Google come IdP del pool di identità, Configurazione di Login with Amazon come IdP di pool di identità e Configurazione di Accedi con Apple come IdP del pool di identità. Con le identità autenticate dagli sviluppatori, puoi registrare e autenticare gli utenti tramite il tuo processo di autenticazione esistente, continuando a utilizzare Amazon Cognito per sincronizzare i dati degli utenti e accedere alle risorse. AWS L'utilizzo di identità autenticate dagli sviluppatori prevede l'interazione tra il dispositivo dell'utente finale, il back-end per l'autenticazione e Amazon Cognito. Per maggiori dettagli, consulta Understanding Amazon Cognito Authentication Part 2: Developer Authenticated Identities nel blog. AWS

Informazioni sul flusso di autenticazione

Il funzionamento dell'GetOpenIdTokenForDeveloperIdentityAPI può avviare l'autenticazione degli sviluppatori sia per l'autenticazione avanzata che per quella di base. Questa API autentica una richiesta con credenziali amministrative. La `Logins` mappa è il nome di un provider di sviluppatori di pool di identità, ad esempio `login.mydevprovider` abbinato a un identificatore personalizzato.

Esempio:

```
"Logins": {
        "login.mydevprovider": "my developer identifier"
    }
```

Autenticazione avanzata

Chiama l'operazione GetCredentialsForIdentityAPI con una `Logins` mappa con il nome `cognito-identity.amazonaws.com` e il valore del token da`GetOpenIdTokenForDeveloperIdentity`.

Esempio:

```
"Logins": {
        "cognito-identity.amazonaws.com": "eyJra12345EXAMPLE"
    }
```

`GetCredentialsForIdentity`con identità autenticate dallo sviluppatore restituisce credenziali temporanee per il ruolo autenticato predefinito del pool di identità.

Autenticazione di base

Chiama l'operazione AssumeRoleWithWebIdentityAPI e richiedi qualsiasi ruolo IAM che abbia una relazione di trust appropriata definita. `RoleArn` Imposta il valore di `WebIdentityToken` per il token ottenuto da`GetOpenIdTokenForDeveloperIdentity`.

Per informazioni sull'authflow delle identità autenticate dagli sviluppatori e su come si differenziano dalle identità dei provider esterni, consulta. Flusso di autenticazione dei pool di identità (identità federate)

Definisci un nome del provider per sviluppatori e associalo con un pool di identità

Per usare le identità autenticate dagli sviluppatori, sarà necessario un pool di identità associato al provider degli sviluppatori. A tale scopo, procedere nel modo seguente:

Per aggiungere un provider degli sviluppatori personalizzato

1. Scegli Pool di identità dalla console di Amazon Cognito. Seleziona un pool di identità.
2. Seleziona la scheda Accesso utente.
3. Seleziona Aggiungi provider di identità.
4. Scegli Provider degli sviluppatori personalizzato.
5. Inserisci un Nome del provider di sviluppatori. Non è possibile modificare o eliminare il provider degli sviluppatori dopo averlo aggiunto.
6. Seleziona Salva modifiche.

Nota: una volta impostato, il nome del provider non può essere modificato.

Per ulteriori istruzioni su come utilizzare la console di Amazon Cognito, consulta Utilizzo della console Amazon Cognito.

Implementa un provider di identità

Android

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende `AWSAbstractCognitoIdentityProvider`. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

Di seguito è riportato un esempio di base di provider di identità.

```
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;
    }

  }
}
```

Per utilizzare questo provider di identità, è necessario passarlo in `CognitoCachingCredentialsProvider`. Ecco un esempio:

```
DeveloperAuthenticationProvider developerProvider = new
 DeveloperAuthenticationProvider( null, "IDENTITYPOOLID", context, Regions.USEAST1);
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider( context, developerProvider, Regions.USEAST1);
```

iOS - objective-C

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende AWSCognitoCredentialsProviderHelper. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

```
@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
```

Per utilizzare questo provider di identità, è necessario passarlo in `AWSCognitoCredentialsProvider`, come illustrato nell'esempio seguente:

```
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];
```

Se intendi supportare sia le identità non autenticate sia le identità autenticate dagli sviluppatori, sostituisci il metodo `logins` nell'implementazione di `AWSCognitoCredentialsProviderHelper`.

```
- (AWSTask<NSDictionary<NSString *, NSString *> *> *)logins {
    if(/*logic to determine if user is unauthenticated*/) {
        return [AWSTask taskWithResult:nil];
    }else{
        return [super logins];
    }
}
```

Se intendi supportare le identità autenticate dagli sviluppatori e i provider social, devi indicare chi è il provider attuale nell'implementazione `logins` di `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

Per utilizzare le identità autenticate dagli sviluppatori, implementa la tua classe di provider di identità che estende AWSCognitoCredentialsProviderHelper. La tua classe di provider di identità dovrebbe restituire un oggetto di risposta contenente il token come attributo.

```
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)
}
```

Per utilizzare questo provider di identità, è necessario passarlo in `AWSCognitoCredentialsProvider`, come illustrato nell'esempio seguente:

```
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
```

Se intendi supportare sia le identità non autenticate sia le identità autenticate dagli sviluppatori, sostituisci il metodo `logins` nell'implementazione di `AWSCognitoCredentialsProviderHelper`.

```
override func logins () -> AWSTask<NSDictionary> {
    if(/*logic to determine if user is unauthenticated*/) {
        return AWSTask(result:nil)
    }else {
        return super.logins()
    }
}
```

Se intendi supportare le identità autenticate dagli sviluppatori e i provider social, devi indicare chi è il provider attuale nell'implementazione `logins` di `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

Una volta ottenuti un ID identità e un token della sessione dal back-end, sarà necessario passarli nel provider `AWS.CognitoIdentityCredentials`. Ecco un esempio:

```
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'
   }
});
```

Unità

Per utilizzare le identità autenticate dagli sviluppatori, è necessario estendere `CognitoAWSCredentials` e sostituire il metodo `RefreshIdentity` per recuperare l'ID identità e il token dell'utente dal back-end, quindi restituirli. Di seguito è riportato un semplice esempio di provider di identità che si mette in contatto con un ipotetico back-end in "esempio.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);
    }
}
```

Il codice in alto utilizza un oggetto dispatcher di thread per chiamare una co-routine. Se non disponi di un modo per eseguire questa operazione nel tuo progetto, puoi utilizzare i seguenti script nelle scene:

```
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

Per utilizzare le identità autenticate dagli sviluppatori, è necessario estendere `CognitoAWSCredentials` e sostituire il metodo `RefreshIdentity` per recuperare l'ID identità e il token dell'utente dal back-end, quindi restituirli. Di seguito è riportato un esempio di base di provider di un identità che si mette in contatto con un ipotetico back-end in "esempio.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;
    }
}
```

Aggiornamento della mappa degli accessi (solo Android e iOS)

Android

Una volta completata correttamente l'autenticazione dell'utente con il sistema di autenticazione, aggiorna la mappa degli accessi con il nome del provider degli sviluppatori e un identificatore utente per gli sviluppatori. Questa è una stringa alfanumerica che identifica in maniera univoca un utente nel sistema di autenticazione. Assicurati di chiamare il metodo `refresh` dopo aver aggiornato la mappa degli accessi, poiché `identityId` potrebbe essere cambiato:

```
HashMap<String, String> loginsMap = new HashMap<String, String>();
loginsMap.put(developerAuthenticationProvider.getProviderName(),
 developerUserIdentifier);

credentialsProvider.setLogins(loginsMap);
credentialsProvider.refresh();
```

iOS - objective-C

L'SDK per iOS chiama solo il metodo `logins` per ottenere le mappe degli accessi più recenti se non sono disponibili le credenziali o se sono scadute. Se vuoi forzare l'SDK a ottenere nuove credenziali (ad esempio, l'utente è passato dall'essere non autenticato a essere autenticato e desideri credenziali rispetto all'utente autenticato), chiama `clearCredentials` su `credentialsProvider`.

```
[credentialsProvider clearCredentials];
```

iOS - Swift

L'SDK per iOS chiama solo il metodo `logins` per ottenere le mappe degli accessi più recenti se non sono disponibili le credenziali o se sono scadute. Se vuoi forzare l'SDK a ottenere nuove credenziali (ad esempio, se il tuo utente è passato dall'essere non autenticato a essere autenticato e vuoi avere delle credenziali dell'utente autenticato), chiama `clearCredentials` sul tuo `credentialsProvider`.

```
credentialsProvider.clearCredentials()
```

Ottenimento di un token (lato server)

È possibile ottenere un [GetOpenIdTokenForDeveloperIdentity]token chiamando. Questa API deve essere richiamata dal backend utilizzando le credenziali AWS dello sviluppatore. Non deve essere invocata dall'SDK client. L'API riceve l'ID pool di identità di Cognito, una mappa degli accessi contenente il nome del provider di identità come chiave e l'identificatore come valore, e, facoltativamente, un ID identità di Cognito (ad esempio, se stai rendendo autenticato un utente non autenticato). L'identificatore può essere il nome utente dell'utente, un indirizzo e-mail o un valore numerico. L'API risponde alla tua chiamata con un ID di Cognito univoco per il tuo utente e con un token Cognito OpenID Connect per l'utente finale.

Qualche consiglio da tenere a mente relativamente al token restituito da `GetOpenIdTokenForDeveloperIdentity`:

- Puoi specificare un periodo di scadenza personalizzato per il token in modo da memorizzarlo nella cache. Se non fornisci alcun periodo di scadenza personalizzato, il token è valido per 15 minuti.
- È possibile impostare un periodo massimo di durata del token di 24 ore.
- Sii consapevole delle implicazioni di sicurezza relative all'aumento della durata del token. Se un utente malintenzionato ottiene questo token, può scambiarlo con AWS credenziali per l'utente finale per la durata del token.

I seguenti frammenti di codice Java mostrano come inizializzare un client Amazon Cognito e recuperare un token per un'identità autenticata dagli sviluppatori.

```
// 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 * 15l);
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
//...
```

Seguendo i passaggi precedenti, sarai in grado di integrare le identità autenticate dagli sviluppatori nell'app. In caso di problemi o domande, non esitare a scrivere sui nostri forum.

Connessione a un'identità social esistente

Quando utilizzi le identità autenticate dagli sviluppatori, tutti i collegamenti dei provider devono essere eseguiti dal back-end. Per collegare un'identità personalizzata all'identità social di un utente (Login with Amazon, Accedi con Apple, Facebook o Google), aggiungi il token del provider di identità alla mappa degli accessi quando GetOpenIdTokenForDeveloperIdentitychiami. Per far sì che ciò accada, quando chiami il back-end dal tuo SDK client per autenticare l'utente finale, trasmetti anche il token del provider del social dell'utente finale.

Ad esempio, se stai cercando di collegare un'identità personalizzata a Facebook, oltre all'identificatore del provider di identità, sarà necessario aggiungere alla mappa degli accessi anche il token di Facebook quando chiami `GetOpenIdTokenForDeveloperIdentity`.

```
logins.put("YOUR_IDENTITY_PROVIDER_NAME","YOUR_END_USER_IDENTIFIER");
logins.put("graph.facebook.com","END_USERS_FACEBOOK_ACCESSTOKEN");
```

Supporto delle transizioni tra provider

Android

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. La differenza principale tra le identità autenticate dagli sviluppatori e altre identità (identità non autenticate e identità autenticate tramite provider pubblici) risiede nella modalità in cui identityId e token vengono ottenuti. Per altre identità, l'applicazione per dispositivi mobili interagisce direttamente con Amazon Cognito anziché contattare il sistema di autenticazione. Pertanto, l'applicazione mobile deve essere in grado di supportare due flussi distinti, a seconda della scelta dell'utente dell'app. Per questo motivo sarà necessario apportare alcune modifiche al provider di identità personalizzato.

Il metodo `refresh` controlla la mappa degli accessi. Se la mappa non è vuota e contiene una chiave con il nome del provider degli sviluppatori, chiama il back-end. Altrimenti, chiama il getIdentityId metodo e restituisci 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;
   }
```

```
}
```

Analogamente il metodo `getIdentityId` dispone di due flussi, in base al contenuto della mappa degli accessi:

```
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

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. Per fare ciò, sovrascrivi il

AWSCognitoCredentialsProviderHelper`logins`metodo per poter restituire la mappa di accesso corretta in base all'attuale provider di identità. Questo esempio illustra come muoversi tra un'identità non autenticata, un'identità autenticata da Facebook e dagli sviluppatori.

```
- (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];
    }
}
```

Quando effettui la transizione da non autenticato ad autenticato, è necessario chiamare `[credentialsProvider clearCredentials];` per forzare l'SDK a ricevere nuove credenziali autenticate. Quando si passa tra due provider autenticati e non si cerca di collegarli (ad esempio, se non vengono forniti token per più provider nel dizionario degli accessi), chiama `[credentialsProvider clearKeychain];`. Questo cancellerà sia le credenziali che le identità e forzerà l'SDK a ottenerne di nuove.

iOS - Swift

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. A tale scopo, sostituisci il AWSCognitoCredentialsProviderHelper`logins`metodo per poter restituire la mappa di accesso corretta in base all'attuale provider di identità. Questo esempio illustra come muoversi tra un'identità non autenticata, un'identità autenticata da Facebook e dagli sviluppatori.

```
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()
    }
}
```

Quando effettui la transizione da non autenticato ad autenticato, è necessario chiamare `credentialsProvider.clearCredentials()` per forzare l'SDK a ricevere nuove credenziali autenticate. Quando alterni tra due provider autenticati e non cerchi di collegarli (ad esempio, se non fornisci token per più provider nel tuo dizionario degli accessi), è necessario chiamare `credentialsProvider.clearKeychain()`. Questo cancellerà sia le credenziali che le identità e forzerà l'SDK a ottenerne di nuove.

Unità

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. La differenza principale tra le identità autenticate dagli sviluppatori e altre identità (identità non autenticate e identità autenticate tramite provider pubblici) risiede nella modalità in cui identityId e token vengono ottenuti. Per altre identità, l'applicazione per dispositivi mobili interagisce direttamente con Amazon Cognito anziché contattare il sistema di autenticazione. L'applicazione per dispositivi mobili deve essere in grado di supportare due flussi distinti, a seconda della scelta dell'utente dell'app. Per questo motivo sarà necessario apportare alcune modifiche al provider di identità personalizzato.

Il modo consigliato per farlo in Unity è estendere il proprio provider di AbstractCognitoIdentityProvider identità AmazonCognitoEnhancedIdentityProvide anziché chiamare il RefreshAsync metodo principale anziché il proprio nel caso in cui l'utente non sia autenticato con il proprio backend. Se l'utente viene autenticato, puoi utilizzare lo stesso flusso spiegato precedentemente.

Xamarin

L'applicazione potrebbe richiedere il supporto di identità non autenticate o di identità autenticate tramite provider pubblici (Login with Amazon, Accedi con Apple, Facebook o Google) oltre alle identità autenticate dagli sviluppatori. La differenza principale tra le identità autenticate dagli sviluppatori e altre identità (identità non autenticate e identità autenticate tramite provider pubblici) risiede nella modalità in cui identityId e token vengono ottenuti. Per altre identità, l'applicazione per dispositivi mobili interagisce direttamente con Amazon Cognito anziché contattare il sistema di autenticazione. L'applicazione per dispositivi mobili deve essere in grado di supportare due flussi distinti, a seconda della scelta dell'utente dell'app. Per questo motivo sarà necessario apportare alcune modifiche al provider di identità personalizzato.

Cambio degli utenti non autenticati in utenti autenticati (pool di identità)

I pool di identità di Amazon Cognito supportano sia utenti autenticati che non autenticati. Gli utenti non autenticati ottengono l'accesso alle risorse AWS anche se non sono connessi con uno dei provider di identità (IdP, Identity Provider). Tale livello di accesso è utile per visualizzare i contenuti agli utenti prima che effettuano l'accesso. Ogni utente non autenticato ha un'identità univoca nel pool di identità, anche se non ha eseguito individualmente l'accesso ed è stato autenticato.

In questa sezione viene descritto il caso in cui l'utente sceglie di cambiare dall'accesso con un'identità non autenticata all'uso di un'identità autenticata.

Android

Gli utenti possono accedere all'app come ospiti non autenticati. Alla fine potrebbero decidere di effettuare l'accesso utilizzando uno dei provider di identità supportati. Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati del profilo vengono uniti automaticamente.

La tua applicazione viene informata di un'unione dei profili tramite l'interfaccia `IdentityChangedListener`. Implementa il metodo `identityChanged` nell'interfaccia per ricevere questi messaggi:

```
@override
public void identityChanged(String oldIdentityId, String newIdentityId) {
    // handle the change
}
```

iOS - Objective-C

Gli utenti possono accedere all'app come ospiti non autenticati. Alla fine potrebbero decidere di effettuare l'accesso utilizzando uno dei provider di identità supportati. Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati dei due profili vengono uniti automaticamente.

`NSNotificationCenter` informa la tua applicazione di un'unione dei profili:

```
[[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

Gli utenti possono accedere all'app come ospiti non autenticati. Alla fine potrebbero decidere di effettuare l'accesso utilizzando uno dei provider di identità supportati. Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati dei due profili vengono uniti automaticamente.

`NSNotificationCenter` informa la tua applicazione di un'unione dei profili:

```
[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

Utente inizialmente non autenticato

Gli utenti in genere iniziano con il ruolo non autenticato. Per questo ruolo, imposti la proprietà delle credenziali dell'oggetto di configurazione senza una proprietà di accesso. In questo caso, la configurazione predefinita potrebbe essere simile alla seguente:

```

// set the default config object
```

```
var creds = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: 'us-east-1:1699ebc0-7900-4099-b910-2df94f52a030'
});
AWS.config.credentials = creds;
```

Cambio a utente autenticato

Quando un utente non autenticato accede a un provider di identità e dispone di un token, puoi cambiare l'utente da non autenticato ad autenticato chiamando una funzione personalizzata che aggiorna l'oggetto credenziali e aggiunge il token di accesso:

```
// 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;
}
```

Inoltre puoi creare un oggetto `CognitoIdentityCredentials`. In tal caso, è necessario reimpostare le proprietà delle credenziali di qualsiasi oggetto esistente del servizio per riflettere le informazioni di configurazione delle credenziali aggiornate. Vedi la sezione relativa all'uso dell'oggetto configurazione globale.

Per ulteriori informazioni sull'oggetto `CognitoIdentityCredentials`, consulta AWS.CognitoIdentityCredentials nella documentazione di riferimento delle API AWS SDK for JavaScript.

Unità

Gli utenti possono accedere all'app come ospiti non autenticati. Alla fine potrebbero decidere di effettuare l'accesso utilizzando uno dei provider di identità supportati. Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati del profilo vengono uniti automaticamente.

Puoi iscriverti a `IdentityChangedEvent` per ricevere notifiche sulle unioni dei profili:

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
 CognitoAWSCredentials.IdentityChangedArgs e)
```

```
{
    // handle the change
    Debug.log("Identity changed from " + e.OldIdentityId + " to " + e.NewIdentityId);
};
```

Xamarin

Gli utenti possono accedere all'app come ospiti non autenticati. Alla fine potrebbero decidere di effettuare l'accesso utilizzando uno dei provider di identità supportati. Amazon Cognito verifica che una vecchia identità usi lo stesso identificatore univoco della nuova identità e i dati del profilo vengono uniti automaticamente.

```
credentialsProvider.IdentityChangedEvent += delegate(object sender,
 CognitoAWSCredentials.IdentityChangedArgs e){
    // handle the change
    Console.WriteLine("Identity changed from " + e.OldIdentityId + " to " +
 e.NewIdentityId);
};
```

Amazon Cognito Sync

> ⚠ Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Sync è un Servizio AWS e una libreria client che rende possibile la sincronizzazione di dati dell'utente relativi all'applicazione tra più dispositivi. Con Amazon Cognito Sync puoi sincronizzare i dati del profilo utente tra dispositivi mobili e il Web, senza utilizzare il tuo back-end. Le librerie client archiviano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati. Se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Per informazioni sulla disponibilità regionale delle identità di Amazon Cognito, consulta Disponibilità delle regioni del servizio AWS.

Per ulteriori informazioni su Amazon Cognito Sync consulta i seguenti argomenti.

Argomenti

- Nozioni di base su Amazon Cognito Sync
- Sincronizzazione dei dati
- Gestione dei callback
- Sincronizzazione push
- Amazon Cognito Streams
- Amazon Cognito Events

Nozioni di base su Amazon Cognito Sync

> ⚠ Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Sync è un servizio AWS e una libreria client che consente la sincronizzazione tra più dispositivi di dati dell'utente relativi all'applicazione. Puoi usarlo per sincronizzare i dati del profilo utente tra dispositivi mobili e i Web. Le librerie client memorizzano localmente i dati nella cache, in modo che la tua app sia in grado di leggere e scrivere i dati indipendentemente dallo stato di connettività del dispositivo. Quando il dispositivo è online, puoi sincronizzare i dati e, se configuri la sincronizzazione push, puoi avvisare immediatamente altri dispositivi della disponibilità di un aggiornamento.

Configurazione di un pool di identità in Amazon Cognito

Amazon Cognito Sync richiede un pool di identità Amazon Cognito per fornire le identità degli utenti. Prima di utilizzare Amazon Cognito Sync, dovrai prima configurare un pool di identità. Per creare un pool di identità e installare l'SDK, consulta Guida introduttiva ai pool di identità di Amazon Cognito.

Archiviazione e sincronizzazione dei dati

Dopo aver configurato il tuo pool di identità e installato l'SDK, puoi iniziare ad archiviare e a sincronizzare dati tra i dispositivi. Per ulteriori informazioni, consulta Sincronizzazione dei dati.

Sincronizzazione dei dati

> ⚠ Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.

Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito ti consente di salvare i dati dell'utente finale in set di dati contenenti coppie chiave-valore. Questi dati sono associati a un'identità di Amazon Cognito nel tuo pool di identità, in modo che sia possibile accedervi attraverso gli accessi e i dispositivi. Per sincronizzare i dati tra il servizio Amazon Cognito e i dispositivi di un utente finale, richiama il metodo di sincronizzazione. Ogni set di dati può avere una dimensione massima di 1 MB. Puoi associare fino a un massimo di 20 set di dati con un'identità.

Il client Amazon Cognito Sync crea una cache locale per i dati di identità. Quando legge e scrive chiavi, la tua app comunica con questa cache locale. Questo garantisce che tutte le modifiche che effettui sul dispositivo siano immediatamente disponibili sullo stesso, anche quando sei offline. Quando viene chiamato il metodo di sincronizzazione, le modifiche del servizio vengono trasferite al dispositivo e eventuali modifiche locali vengono trasferite al servizio. A questo punto, le modifiche sono disponibili per la sincronizzazione con altri dispositivi.

Inizializzazione del client di Amazon Cognito Sync

Per inizializzare il client di Amazon Cognito Sync, è necessario creare prima un fornitore di credenziali. Il fornitore di credenziali acquisisce le credenziali AWS temporanee per permettere alla tua app di accedere alle tue risorse AWS. Devi anche importare i file di intestazione necessari. Utilizza la procedura seguente per inizializzare il client di Amazon Cognito Sync.

Android

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa il pacchetto Amazon Cognito come segue: `import com.amazonaws.mobileconnectors.cognito.*;`
3. Inizializza Amazon Cognito Sync. Passa il contesto dell'app Android, l'ID del pool di identità, una Regione AWS e un fornitore di credenziali Amazon Cognito inizializzato come segue:

```
CognitoSyncManager client = new CognitoSyncManager(
    getApplicationContext(),
    Regions.YOUR_REGION,
```

```
    credentialsProvider);
```

iOS - Objective-C

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa `AWSCore` e `Cognito` e inizializza `AWSCognito` come segue:

```
#import <AWSiOSSDKv2/AWSCore.h>
#import <AWSCognitoSync/Cognito.h>

AWSCognito *syncClient = [AWSCognito defaultCognito];
```

3. Se usi CocoaPods, sostituisci `<AWSiOSSDKv2/AWSCore.h>` con `AWSCore.h`. Segui la stessa sintassi per l'importazione di Amazon Cognito.

iOS - Swift

1. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
2. Importa e inizializza `AWSCognito` come segue:

```
import AWSCognito
let syncClient = AWSCognito.default()!
```

JavaScript

1. Scarica Amazon Cognito Sync Manager for JavaScript.
2. Includi la libreria del manager della sincronizzazione nel tuo progetto;
3. Crea un fornitore di credenziali, seguendo le istruzioni in Ottenere le credenziali.
4. Inizializza il manager della sincronizzazione come segue:

```
var syncManager = new AWS.CognitoSyncManager();
```

Unità

1. Avrai bisogno inizialmente di creare un'istanza delle `CognitoAWSCredentials`, seguendo le istruzioni in Ottenere le credenziali.
2. Creare un'istanza di `CognitoSyncManager`. Passa l'oggetto delle `CognitoAwsCredentials` e un `AmazonCognitoSyncConfig` e includi almeno il set Regione, come segue:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Xamarin

1. Crea un'istanza delle `CognitoAWSCredentials`, seguendo le istruzioni in Ottenere le credenziali.
2. Creare un'istanza di `CognitoSyncManager`. Passa l'oggetto delle `CognitoAwsCredentials` e un `AmazonCognitoSyncConfig` e includi almeno il set Regione, come segue:

```
AmazonCognitoSyncConfig clientConfig = new AmazonCognitoSyncConfig { RegionEndpoint =
 REGION };
CognitoSyncManager syncManager = new CognitoSyncManager(credentials, clientConfig);
```

Comprendere i set di dati

Amazon Cognito organizza i dati del profilo dell'utente finale in set di dati. Ogni set di dati può contenere fino a 1 MB di dati sotto forma di coppie chiave-valore. Un set di dati è l'entità più granulare che puoi sincronizzare. Le operazioni di lettura e scrittura eseguite su un set di dati riguardano solo l'archivio locale fino a quando il metodo di sincronizzazione viene invocato. Amazon Cognito identifica un set di dati attraverso una stringa univoca. Puoi creare un nuovo set di dati oppure aprirne uno esistente nel modo seguente.

Android

```
Dataset dataset = client.openOrCreateDataset("datasetname");
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
dataset.delete();
dataset.synchronize(syncCallback);
```

iOS - Objective-C

```
AWSCognitoDataset *dataset = [syncClient openOrCreateDataset:@"myDataSet"];
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
[dataset clear];
[dataset synchronize];
```

iOS - Swift

```
let dataset = syncClient.openOrCreateDataset("myDataSet")!
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito:

```
dataset.clear()
dataset.synchronize()
```

JavaScript

```
syncManager.openOrCreateDataset('myDatasetName', function(err, dataset) {
   // ...
});
```

Unità

```
string myValue = dataset.Get("myKey");
```

```
dataset.Put("myKey", "newValue");
```

Per eliminare una chiave dal set di dati, utilizza `Remove` come segue:

```
dataset.Remove("myKey");
```

Xamarin

```
Dataset dataset = syncManager.OpenOrCreateDataset("myDatasetName");
```

Per eliminare un set di dati, per prima cosa chiama il metodo per rimuoverlo dallo spazio di archiviazione locale, quindi richiama il metodo `synchronize` per eliminare il set di dati da Amazon Cognito nel modo seguente:

```
dataset.Delete();
dataset.SynchronizeAsync();
```

Lettura e scrittura dei dati nei set di dati

Funzione di set di dati di Amazon Cognito come dizionari, con i valori accessibili tramite chiave. Le chiavi e valori di un set di dati possono essere letti, aggiunti o modificati esattamente come se il set di dati fosse un dizionario, come mostrato negli esempi seguenti.

Nota che i valori che scrivi su un set di dati locali riguardano solo la copia memorizzata nella cache locale finché non chiami il metodo di sincronizzazione.

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);
});
```

Unità

```
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

Puoi utilizzare il metodo `remove` per rimuovere le chiavi da un set di dati come segue:

```
dataset.remove("myKey");
```

iOS - Objective-C

Per eliminare una chiave dal set di dati, utilizza `removeObjectForKey` come segue:

```
[dataset removeObjectForKey:@"myKey"];
```

iOS - Swift

Per eliminare una chiave dal set di dati, utilizza `removeObjectForKey` come segue:

```
dataset.removeObjectForKey("myKey")
```

Unità

Per eliminare una chiave dal set di dati, utilizza `Remove` come segue:

```
dataset.Remove("myKey");
```

Xamarin

Usa il comando `Remove` per eliminare una chiave dal set di dati:

```
dataset.Remove("myKey");
```

Sincronizzazione dei dati locali con lo store di sincronizzazione

Android

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.synchronize(syncCallback);
```

Il metodo `synchronize` riceve un'implementazione dell' interfaccia `SyncCallback`, illustrato di seguito.

Il metodo `synchronizeOnConnectivity()` tenta di sincronizzare quando è disponibile la connettività. Se la connettività è immediatamente disponibile, `synchronizeOnConnectivity()` si comporta come `synchronize()`. In caso contrario, monitora i cambiamenti di connettività ed esegue una sincronizzazione una volta che la connettività è disponibile. Se `synchronizeOnConnectivity()`è chiamata più volte, solo l'ultima richiesta di sincronizzazione viene mantenuta, e solo l'ultima callback verrà attivata. Se il set di dati o la callback vengono sottoposti al processo di garbage collection, questo metodo non eseguirà una sincronizzazione e la callback non verrà attivata.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

iOS - Objective-C

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

Il metodo `synchronize` è asincrono e restituisce un oggetto `AWSTask` per gestire la risposta:

```
[[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;
}];
```

Il metodo `synchronizeOnConnectivity` tenta di sincronizzare quando il dispositivo ha la connettività. In primo luogo, `synchronizeOnConnectivity` verifica la connettività e se il dispositivo è online immediatamente invoca la sincronizzazione e restituisce l'oggetto `AWSTask` associato al tentativo.

Se il dispositivo è offline, `synchronizeOnConnectivity` 1) pianifica un sincronizzare per la prossima volta che il dispositivo sarà online e 2) restituisce un `AWSTask` con risultato nullo. La sincronizzazione programmata è valida solo per il ciclo di vita dell'oggetto del set di dati. I dati non verranno sincronizzati se l'app viene chiusa prima che la connettività sia riconquistata. Se desideri essere avvisato ogni qualvolta hanno luogo eventi durante la sincronizzazione programmata, è necessario aggiungere osservatori delle notifiche trovati in `AWSCognito`.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

iOS - Swift

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

Il metodo `synchronize` è asincrono e restituisce un oggetto `AWSTask` per gestire la risposta:

```
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
})
```

Il metodo `synchronizeOnConnectivity` tenta di sincronizzare quando il dispositivo ha la connettività. In primo luogo, `synchronizeOnConnectivity` verifica la connettività e se il dispositivo è online immediatamente invoca la `synchronize` e restituisce l'oggetto `AWSTask` associato al tentativo.

Se il dispositivo è offline, `synchronizeOnConnectivity` 1) pianifica un sincronizzare per la prossima volta che il dispositivo sarà online e 2) restituisce un oggetto `AWSTask` con un risultato nullo. La sincronizzazione programmata è valida solo per il ciclo di vita dell'oggetto del set di dati. I

dati non verranno sincronizzati se l'app viene chiusa prima che la connettività sia riconquistata. Se desideri essere avvisato ogni qualvolta hanno luogo eventi durante la sincronizzazione programmata, è necessario aggiungere osservatori delle notifiche trovati in `AWSCognito`.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

JavaScript

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.synchronize();
```

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

Unità

Il metodo di sincronizzazione confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.Synchronize();
```

La sincronizzazione genererà in modo asincrono e terminerà chiamando una delle diverse callback che puoi specificare nel set di dati.

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

Xamarin

Il metodo `synchronize` confronta i dati memorizzati nella cache locale con i dati archiviati nello store di Amazon Cognito Sync. Le modifiche remote vengono estratte dallo store Amazon

Cognito Sync; la risoluzione dei conflitti viene richiamata se si verificano conflitti e i valori aggiornati sul dispositivo vengono inviati al servizio. Per sincronizzare il set di dati, chiama il suo metodo `synchronize`:

```
dataset.SynchronizeAsync();
```

Per ulteriori informazioni sulla sincronizzazione del set di dati e le diverse callback, consulta Gestione dei callback.

Gestione dei callback

> **Warning**
> Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Questa sezione descrive come gestire le callback.

Android

Interfaccia SyncCallback

Se implementi l'interfaccia `SyncCallback`, puoi ricevere notifiche sulla tua app riguardanti la sincronizzazione del set di dati. La tua app può quindi prendere decisioni attive sull'eliminazione di dati locali, sull'unione di profili autenticati e non autenticati e la risoluzione di conflitti di sincronizzazione. È consigliabile implementare i seguenti metodi, che sono richiesti dall'interfaccia:

- `onSuccess()`
- `onFailure()`
- `onConflict()`
- `onDatasetDeleted()`
- `onDatasetsMerged()`

Nota che, se non desideri specificare tutti le callback, puoi anche utilizzare la classe `DefaultSyncCallback`, che fornisce implementazioni predefinite e vuote per ognuna di esse.

onSuccess

La callback `onSuccess()` viene attivata quando un set di dati è stato scaricato con successo dall'archivio di sincronizzazione.

```
@Override
public void onSuccess(Dataset dataset, List<Record> newRecords) {
}
```

onFailure

onFailure() viene chiamata se si verifica un'eccezione durante la sincronizzazione.

```
@Override
public void onFailure(DataStorageException dse) {
}
```

onConflict

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. Il metodo `onConflict()` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, il client Amazon Cognito Sync utilizza per impostazione predefinita la modifica più recente.

```
@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

Quando un set di dati viene eliminato, il client Amazon Cognito usa l'interfaccia `SyncCallback` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Implementa il metodo `onDatasetDeleted()` per comunicare all'SDK del client che cosa fare con i dati locali.

```
@Override
public boolean onDatasetDeleted(Dataset dataset, String datasetName) {
    // return true to delete the local copy of the dataset
    return true;
}
```

onDatasetMerged

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite il metodo `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

Notifiche di sincronizzazione

Il client Amazon Cognito durante una chiamata di sincronizzazione emetterà un numero di eventi di `NSNotification`. Puoi registrarti per monitorare queste notifiche attraverso lo standard `NSNotificationCenter`:

```
[NSNotificationCenter defaultCenter]
```

```
    addObserver:self
    selector:@selector(myNotificationHandler:)
    name:NOTIFICATION_TYPE
    object:nil];
```

Amazon Cognito supporta cinque tipi di notifica, elencati di seguito.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione è in fase di avvio. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione viene completata (correttamente o meno). La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato.

AWSCognitoDidFailToSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione non va a buon fine. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e l'errore chiave, che conterrà l'errore che ha causato il fallimento.

AWSCognitoDidChangeRemoteValueNotification

Chiamato quando le modifiche locali vengono inviate correttamente ad Amazon Cognito. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e le chiavi chiave, che conterranno un NSArray di chiavi record che sono state trasmesse.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Viene chiamato quando un valore locale si modifica a causa di un'operazione di sincronizzazione. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e le chiavi principali, che conterranno un NSArray di chiavi record modificate.

Gestore della risoluzione del conflitto

Durante un'operazione di sincronizzazione, possono sorgere dei conflitti se la stessa chiave è stata modificata nell'archivio locale e in quello di sincronizzazione. Se non hai impostato un gestore della risoluzione dei conflitti, Amazon Cognito sceglierà di default l'aggiornamento più recente.

Con l'implementazione e l'assegnazione di un AWSCognitoRecordConflictHandler puoi modificare la risoluzione dei conflitti predefiniti. Il conflitto del parametro di input AWSCognitoConflict contiene un oggetto AWSCognitoRecord sia per i dati memorizzati nella cache locale, sia per i record di conflitto nello store di sincronizzazione. Se usi AWSCognitoConflict puoi risolvere il conflitto con il record locale: [conflitto resolveWithLocalRecord], il record remoto: [conflitto resolveWithRemoteRecord] o con il valore di un nuovo marchio: [conflitto resolveWithValue: valore]. Il ritorno a zero da questo metodo impedisce alla sincronizzazione di continuare e il conflitto verrà presentato ancora, la prossima volta che il processo di sincronizzazione si riavvierà.

Puoi impostare il gestore della risoluzione del conflitto al livello del client:

```
client.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
    // always choose local changes
    return [conflict resolveWithLocalRecord];
};
```

Oppure a livello del set di dati:

```
dataset.conflictHandler = ^AWSCognitoResolvedConflict* (NSString *datasetName,
 AWSCognitoConflict *conflict) {
    // override and always choose remote changes
    return [conflict resolveWithRemoteRecord];
};
```

Gestore dell'eliminazione del set di dati

Quando un set di dati viene eliminato, il client Amazon Cognito usa `AWSCognitoDatasetDeletedHandler` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Se non è stato implementato alcun `AWSCognitoDatasetDeletedHandler`, i dati locali saranno rimossi automaticamente. Implementa un `AWSCognitoDatasetDeletedHandler` se desideri mantenere una copia dei dati locali prima della cancellazione, o mantenere i dati locali.

Puoi impostare il gestore dell'eliminazione del set di dati a livello del client:

```
client.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // make a backup of the data if you choose
    ...
    // delete the local data (default behavior)
```

```
    return YES;
};
```

Oppure a livello del set di dati:

```
dataset.datasetDeletedHandler = ^BOOL (NSString *datasetName) {
    // override default and keep the local data
    return NO;
};
```

Gestore dell'unione del set di dati

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le i vengono avvisate dell'unione tramite `DatasetMergeHandler`. Il gestore riceverà il nome del set di dati radice, nonché una vasta gamma di nomi di set di dati contrassegnati come unioni dei set di dati root.

In caso contrario, `DatasetMergeHandler` viene implementato, questi set di dati saranno ignorati ma continueranno a consumare lo spazio nei set di dati con un totale di massimo 20 identità.

Puoi impostare il gestore dell'unione del set di dati a livello del 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];
    }
};
```

Oppure a livello del set di dati:

```
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

Notifiche di sincronizzazione

Il client Amazon Cognito durante una chiamata di sincronizzazione emetterà un numero di eventi di `NSNotification`. Puoi registrarti per monitorare queste notifiche attraverso lo standard `NSNotificationCenter`:

```
NSNotificationCenter.defaultCenter().addObserver(observer: self,
   selector: "myNotificationHandler",
   name:NOTIFICATION_TYPE,
   object:nil)
```

Amazon Cognito supporta cinque tipi di notifica, elencati di seguito.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione è in fase di avvio. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato.

AWSCognitoDidStartSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione viene completata (correttamente o meno). La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato.

AWSCognitoDidFailToSynchronizeNotification

Viene chiamato quando un'operazione di sincronizzazione non va a buon fine. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e l'errore chiave, che conterrà l'errore che ha causato il fallimento.

AWSCognitoDidChangeRemoteValueNotification

Chiamato quando le modifiche locali vengono inviate correttamente ad Amazon Cognito. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e le chiavi chiave, che conterranno un NSArray di chiavi record che sono state trasmesse.

AWSCognitoDidChangeLocalValueFromRemoteNotification

Viene chiamato quando un valore locale si modifica a causa di un'operazione di sincronizzazione. La `userInfo` conterrà il set di dati chiave, che è il nome del set di dati che viene sincronizzato, e le chiavi principali, che conterranno un NSArray di chiavi record modificate.

Gestore della risoluzione del conflitto

Durante un'operazione di sincronizzazione, possono sorgere dei conflitti se la stessa chiave è stata modificata nell'archivio locale e in quello di sincronizzazione. Se non hai impostato un gestore della risoluzione dei conflitti, Amazon Cognito sceglierà per impostazione predefinita l'aggiornamento più recente.

Con l'implementazione e l'assegnazione di un `AWSCognitoRecordConflictHandler` puoi modificare la risoluzione dei conflitti predefiniti. Il conflitto del parametro di input `AWSCognitoConflict` contiene un oggetto `AWSCognitoRecord` sia per i dati memorizzati nella cache locale, sia per i record di conflitto nello store di sincronizzazione. Se usi `AWSCognitoConflict` puoi risolvere il conflitto con il record locale: [conflitto resolveWithLocalRecord], il record remoto: [conflitto resolveWithRemoteRecord] o con il valore di un nuovo marchio: [conflitto resolveWithValue: valore]. Il ritorno a zero da questo metodo impedisce alla sincronizzazione di continuare e il conflitto verrà presentato ancora, la prossima volta che il processo di sincronizzazione si riavvierà.

Puoi impostare il gestore della risoluzione del conflitto al livello del client:

```
client.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Oppure a livello del set di dati:

```
dataset.conflictHandler = {
    (datasetName: String?, conflict: AWSCognitoConflict?) ->
 AWSCognitoResolvedConflict? in
    return conflict.resolveWithLocalRecord()
}
```

Gestore dell'eliminazione del set di dati

Quando un set di dati viene eliminato, il client Amazon Cognito usa `AWSCognitoDatasetDeletedHandler` per confermare se la copia del set di dati memorizzata nella cache locale deve anch'essa essere eliminata. Se non è stato implementato alcun `AWSCognitoDatasetDeletedHandler`, i dati locali saranno rimossi automaticamente. Implementa un `AWSCognitoDatasetDeletedHandler` se desideri mantenere una copia dei dati locali prima della cancellazione, o mantenere i dati locali.

Puoi impostare il gestore dell'eliminazione del set di dati a livello del client:

```
client.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
      return true
}
```

Oppure a livello del set di dati:

```
dataset.datasetDeletedHandler = {
      (datasetName: String!) -> Bool in
      // make a backup of the data if you choose
      ...
      // delete the local data (default behaviour)
      return true
}
```

Handler dell'unione dei set di dati

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le i vengono avvisate dell'unione tramite `DatasetMergeHandler`. Il gestore riceverà il nome del set di dati radice, nonché una vasta gamma di nomi di set di dati contrassegnati come unioni dei set di dati root.

In caso contrario, `DatasetMergeHandler` viene implementato, questi set di dati saranno ignorati ma continueranno a consumare lo spazio nei set di dati con un totale di massimo 20 identità.

Puoi impostare il gestore dell'unione del set di dati a livello del 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()
        }
    }
}
```

Oppure a livello del set di dati:

```
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

Callback di sincronizzazione

Quando esegui una sincronizzazione() su un set di dati, puoi, a tua discrezione, specificare le callback per gestire ciascuno dei seguenti stati:

```
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()

La callback `onSuccess()` viene attivata quando un set di dati è stato aggiornato con successo dallo store di sincronizzazione. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
onSuccess: function(dataset, newRecords) {
   console.log('Successfully synchronized ' + newRecords.length + ' new records.');
}
```

onFailure()

`onFailure()` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
onFailure: function(err) {
   console.log('Synchronization failed.');
   console.log(err);
}
```

onConflict()

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. Il metodo `onConflict()` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
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()

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `onDatasetDeleted()` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Di default, il set di dati non verrà eliminato.

```
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()

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `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);

}
```

Unità

Dopo che hai aperto o creato un set di dati, puoi impostare su questo diverse callback che vengono attivate quando si utilizza il metodo Synchronize. Questo è il modo per registrarvi le tue callback:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Nota che `SyncSuccess` e `SyncFailure` usano += invece di = pertanto puoi sottoscrivervi più di una callback.

OnSyncSuccess

La callback `OnSyncSuccess` viene attivata quando un set di dati è stato aggiornato correttamente dal cloud. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEvent e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
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

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. La callback `OnSyncConflict` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
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

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `OnDatasetDeleted` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Di default, il set di dati non verrà eliminato.

```
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

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `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

Dopo che hai aperto o creato un set di dati, puoi impostare su questo diverse callback che vengono attivate quando si utilizza il metodo Synchronize. Questo è il modo per registrarvi le tue callback:

```
dataset.OnSyncSuccess += this.HandleSyncSuccess;
dataset.OnSyncFailure += this.HandleSyncFailure;
dataset.OnSyncConflict = this.HandleSyncConflict;
dataset.OnDatasetMerged = this.HandleDatasetMerged;
dataset.OnDatasetDeleted = this.HandleDatasetDeleted;
```

Nota che `SyncSuccess` e `SyncFailure` usano **+=** invece di **=** pertanto puoi sottoscrivervi più di una callback.

OnSyncSuccess

La callback `OnSyncSuccess` viene attivata quando un set di dati è stato aggiornato correttamente dal cloud. Se definisci una callback, la sincronizzazione riuscirà automaticamente.

```
private void HandleSyncSuccess(object sender, SyncSuccessEventArgs e)
{
    // Continue with your game flow, display the loaded data, etc.
}
```

OnSyncFailure

`OnSyncFailure` viene chiamata se si verifica un'eccezione durante la sincronizzazione. Se non definisci una callback, la sincronizzazione fallirà automaticamente.

```
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

Possono verificarsi dei conflitti se la stessa chiave è stata modificata nello store locale e in quello di sincronizzazione. La callback `OnSyncConflict` gestisce la risoluzione dei conflitti. Se non implementi questo metodo, la sincronizzazione verrà interrotta ogni volta che si verifica un conflitto.

```
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

Quando un set di dati viene eliminato, il client Amazon Cognito usa la richiamata `OnDatasetDeleted` per decidere se anche la copia del set di dati memorizzato nella cache locale deve essere eliminata. Di default, il set di dati non verrà eliminato.

```
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

Quando due identità precedentemente scollegate vengono collegate, tutti i loro set di dati vengono uniti. Le applicazioni vengono avvisate dell'unione tramite la chiamata `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;
}
```

Sincronizzazione push

> ⚠ Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync. Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito monitora automaticamente l'associazione tra identità e dispositivi. L'uso della funzione della sincronizzazione push ti permette di assicurare che tutte le istanze di una determinata identità siano avvisate della modifica di alcuni dati di identità. La sincronizzazione push garantisce

che, ogni volta che i dati dell'archivio di sincronizzazione cambiano per una determinata identità, tutti i dispositivi associati a tale identità ricevono una notifica push silenziosa che li informa delle modifiche.

> **Note**
>
> La sincronizzazione push non è supportata per JavaScript, Unity o Xamarin.

Prima di poter utilizzare la sincronizzazione push, devi abilitarla nella console Amazon Cognito e configurare l'account per la sincronizzazione.

Creazione di un'app Amazon Simple Notification Service (Amazon SNS)

Crea e configura un'app Amazon SNS per la tua piattaforma supportata, come descritto nella Guida per gli sviluppatori di SNS.

Abilitazione della sincronizzazione push nella console di Amazon Cognito

Puoi abilitare la sincronizzazione push tramite la console Amazon Cognito. Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri attivare la sincronizzazione push. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Manage Identity Pools (Gestisci pool di identità). Viene visualizzata la pagina Federated Identities (Identità federate).
3. Scorri verso il basso e fai clic su Push synchronization (Sincronizzazione push) per ingrandire questa opzione.
4. Nel menu a discesa Service role (Ruolo di servizio) seleziona il ruolo IAM che garantisce a Cognito l'autorizzazione per inviare una notifica SNS. Fai clic su Create role (Crea ruolo) per creare o modificare i ruoli associati al tuo pool di identità nella Console AWS IAM.
5. Seleziona un'applicazione della piattaforma, quindi fai clic su Save Changes (Salva le modifiche).
6. Concedi l'accesso SNS alla tua applicazione

Nella console AWS Identity and Access Management, configura i tuoi ruoli IAM per avere pieno accesso ad Amazon SNS, oppure crea un nuovo ruolo che abbia pieno accesso ad Amazon SNS. L'esempio seguente di policy di attendibilità dei ruoli garantisce ad Amazon Cognito Sync una

capacità limitata di assumere un ruolo IAM. Amazon Cognito Sync può assumere il ruolo solo quando lo fa per conto sia del pool di identità nella condizione `aws:SourceArn` che dell'account nella condizione `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"
                }
            }
        }
    ]
}
```

Per ulteriori informazioni sui ruoli IAM, consulta i Ruoli (Delega e Federazione).

Uso della sincronizzazione push nella tua app: Android

La tua e avrà bisogno di importare i servizi Google Play. Puoi scaricare la versione più recente di SDK di Google Play tramite il gestore SDK di Android. Segui la documentazione di Android suAndroid Implementation per registrare la tua app e ricevere un ID di registrazione da GCM. Una volta che hai ottenuto l'ID di registrazione, hai bisogno di registrare il dispositivo con Amazon Cognito come illustrato nel frammento che segue:

```
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);
}
```

Ora puoi sottoscrivere un dispositivo per ricevere gli aggiornamenti provenienti da un determinato set di dati:

```
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);
    }
}
```

Per interrompere la ricezione di notifiche push da un set di dati, è sufficiente chiamare il metodo di annullamento. Per effettuare la sotttoscrizione a tutti i set di dati (o un sottoinsieme specifico) nell'oggetto `CognitoSyncManager`, usa `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);
    }
}
```

Nella tua implementazione dell'oggetto di AndroidBroadcastReceiver, puoi verificare la versione più recente del set di dati modificati e decidere se la tua app deve sincronizzarsi di nuovo:

```
@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() {
            // ...
        });
    }

}
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile per la sicurezza della chiamata openOrCreateDataset.
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Uso della sincronizzazione push nella tua app: iOS - Objective-C

Per ottenere un token di dispositivo per la tua app, segui la documentazione Apple sulla Registrazione per notifiche remote. Una volta che hai ricevuto il token di dispositivo come un oggetto

NSData dagli APN, è necessario registrare il dispositivo con Amazon Cognito utilizzando il metodo `registerDevice:` del client di sincronizzazione, come illustrato di seguito:

```
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;
      }
    ];
```

In modalità di debug, il tuo dispositivo registrerà con il sandbox degli APN; in modalità di rilascio, registrerà con gli APN. Per ricevere aggiornamenti da un particolare set di dati, usa il metodo `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;
      }
    ];
```

Per interrompere la ricezione di notifiche push da un set di dati, è sufficiente chiamare il metodo `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;
      }
```

```
    ];
```

Per effettuare la sottoscrizione a tutti i set di dati nell'oggetto `AWSCognito`, chiama `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;
      }
    ];
```

Prima di chiamare `subscribeAll`, assicurati di sincronizzare tutti i set di dati almeno una volta, in modo tale che gli stessi set di dati esistano nel server.

Per rispondere alle notifiche push, hai bisogno di implementare il metodo `didReceiveRemoteNotification` nel tuo delegato dell'app:

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification:
(NSDictionary *)userInfo
    {
        [[NSNotificationCenter defaultCenter]
 postNotificationName:@"CognitoPushNotification" object:userInfo];
    }
```

Se pubblichi una notizia tramite il gestore delle notifiche, puoi quindi rispondere alla notifica in qualsiasi punto dell'applicazione in cui disponi di un gestore per il set di dati. Se effettui la sottoscrizione alla notifica in questo modo ...

```
[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(didReceivePushSync:)
    name: :@"CognitoPushNotification" object:nil];
```

... puoi intervenire sulla notifica in questo modo:

```
- (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;
            }];
        }
    }
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile per la sicurezza della chiamata `openOrCreateDataset`.
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Uso della sincronizzazione push nella tua app: iOS - Swift

Per ottenere un token di dispositivo per la tua app, segui la documentazione Apple sulla Registrazione per notifiche remote. Una volta che hai ricevuto il token di dispositivo come un oggetto NSData dagli APN, è necessario registrare il dispositivo con Amazon Cognito utilizzando il registerDevice: del client di sincronizzazione, come illustrato di seguito:

```
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
})
```

In modalità di debug, il tuo dispositivo registrerà con il sandbox degli APN; in modalità di rilascio, registrerà con gli APN. Per ricevere aggiornamenti da un particolare set di dati, usa il metodo `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
})
```

Per interrompere la ricezione di notifiche push da un set di dati, chiama il metodo `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
})
```

Per effettuare la sottoscrizione a tutti i set di dati nell'oggetto `AWSCognito`, chiama `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
})
```

Prima di chiamare `subscribeAll`, assicurati di sincronizzare tutti i set di dati almeno una volta, in modo tale che gli stessi set di dati esistano nel server.

Per rispondere alle notifiche push, hai bisogno di implementare il metodo `didReceiveRemoteNotification` nel tuo delegato dell'app:

```
func application(application: UIApplication, didReceiveRemoteNotification userInfo:
 [NSObject : AnyObject],
  fetchCompletionHandler completionHandler: (UIBackgroundFetchResult) -> Void) {

 NSNotificationCenter.defaultCenter().postNotificationName("CognitoPushNotification",
 object: userInfo)
})
```

Se pubblichi una notizia tramite il gestore delle notifiche, puoi quindi rispondere alla notifica in qualsiasi punto dell'applicazione in cui disponi di un gestore per il set di dati. Se effettui la sottoscrizione alla notifica in questo modo ...

```
NSNotificationCenter.defaultCenter().addObserver(observer:self,
   selector:"didReceivePushSync:",
   name:"CognitoPushNotification",
   object:nil)
```

... puoi intervenire sulla notifica in questo modo:

```
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
          }
```

```
        }
    }
}
```

Le seguenti chiavi sono disponibili nel payload delle notifiche push:

- `source`: cognito-sync. Questo può servire come un fattore di differenziazione tra le notifiche.
- `identityPoolId`: ID pool di identità. Questo può essere utilizzato per la convalida o per informazioni aggiuntive, ma non è integrante dal punto di vista del ricevitore.
- `identityId`: ID identità all'interno del pool.
- `datasetName`: Nome del set di dati che è stato aggiornato. Questo è disponibile per la sicurezza della chiamata `openOrCreateDataset`.
- `syncCount`: Numero di sincronizzazione per il set di dati remoto. Puoi usare questo come un modo per assicurarti che il set di dati locali sia obsoleto e che la sincronizzazione in entrata sia nuova.

Amazon Cognito Streams

> **Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync.** Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Streams offre agli sviluppatori il controllo e l'analisi dei loro dati archiviati in Amazon Cognito. Gli sviluppatori possono ora configurare un flusso Kinesis per ricevere eventi non appena i dati sono aggiornati e sincronizzati. Amazon Cognito può eseguire il push di ogni modifica del set di dati in un flusso Kinesis in tempo reale.

Utilizzando Amazon Cognito Streams, puoi spostare tutti i tuoi dati di Sync in Kinesis, che possono quindi essere distribuiti in uno strumento data warehouse come Amazon Redshift per un'ulteriore analisi. Per ulteriori informazioni su Kinesis, consulta Nozioni di base per l'uso di Amazon Kinesis.

Configurazione dei flussi

Puoi configurare Amazon Cognito Streams nella console Amazon Cognito. Per abilitare Amazon Cognito Streams nella console Amazon Cognito, devi selezionare il flusso Kinesis in cui pubblicare e un ruolo IAM, che concede l'autorizzazione di Amazon Cognito per inserire gli eventi nel flusso selezionato.

Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri configurare Amazon Cognito Streams. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Manage Identity Pools (Gestisci pool di identità). Viene visualizzata la pagina di gestione delle identità federate.
3. Scorri verso il basso e fai clic su Cognito Streams (Flussi di Cognito) per espandere l'operazione.
4. Nel menu a discesa Stream name (Nome del flusso), seleziona il nome di un flusso Kinesis esistente. In alternativa, fai clic su Create stream (Crea flusso) per crearne uno, immettendo un nome di flusso e il numero di shard. Per ulteriori informazioni sugli shard e per valutare il numero di shard di cui hai bisogno per il tuo flusso, consulta la Guida per gli sviluppatori di Kinesis.
5. Nel menu a discesa Publish role (Pubblica ruolo), seleziona il ruolo IAM che garantisce l'autorizzazione di Amazon Cognito per pubblicare il tuo flusso. Fai clic su Create role (Crea ruolo) per creare o modificare i ruoli associati al tuo pool di identità nella Console AWS IAM.
6. Nel menu a discesa Stream status (Stato del flusso) seleziona Enabled (Abilitato) per abilitare gli aggiornamenti del flusso. Fai clic su Save Changes (Salva modifiche).

Dopo che hai configurato correttamente i flussi di Amazon Cognito, tutti i successivi aggiornamenti ai set di dati in questo pool di identità saranno inviati al flusso.

Contenuti del flusso

Ogni record inviato al flusso rappresenta una singola sincronizzazione. Ecco l'esempio di un record inviato al flusso:

```
{
    "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
}
```

Per gli aggiornamenti di dimensioni superiori alla dimensione massima di payload di Kinesis, ovvero 1 MB, Amazon Cognito include un URL Amazon S3 prefirmato con il contenuto completo dell'aggiornamento.

Dopo aver configurato i flussi di Amazon Cognito, se elimini il flusso Kinesis o modifichi l'autorizzazione di attendibilità del ruolo in modo che non possa più essere assunto da Amazon Cognito Sync, disabiliterai i flussi di Amazon Cognito. Devi creare nuovamente il flusso Kinesis o correggere il ruolo e quindi riattivare il flusso.

Pubblicazione in blocco

Una volta che hai configurato i flussi di Amazon Cognito potrai eseguire un'operazione di pubblicazione in blocco per i dati esistenti nel tuo pool di identità. Una volta che avvii un'operazione di pubblicazione in blocco, tramite la console o direttamente tramite l'API, Amazon Cognito inizierà a pubblicare questi dati nello stesso flusso che riceve i tuoi aggiornamenti.

Amazon Cognito non garantisce l'univocità dei dati inviati al flusso quando si utilizza l'operazione di pubblicazione in blocco. Potresti ricevere lo stesso aggiornamento sia come un aggiornamento, sia come parte di una pubblicazione in blocco. Tieni presente questa possibilità durante l'elaborazione del record proveniente dal tuo flusso.

Per pubblicare in blocco tutti i tuoi flussi, segui i passaggi 1-6 della sezione di configurazione dei flussi e quindi fai clic su Start bulk publish (Inizia la pubblicazione in blocco). Hai un limite per un'operazione di pubblicazione in blocco in corso in qualsiasi momento e per una richiesta di pubblicazione in blocco riuscita ogni 24 ore.

Amazon Cognito Events

> **Se non hai mai usato Amazon Cognito Sync, utilizza AWS AppSync.** Come Amazon Cognito Sync, AWS AppSync è un servizio che consente la sincronizzazione dei dati delle applicazioni tra più dispositivi.
> Consente di sincronizzare i dati dell'utente come le preferenze dell'app o lo stato del gioco. Inoltre estende queste funzionalità consentendo a più utenti di sincronizzare e collaborare in tempo reale su dati condivisi.

Amazon Cognito Events ti consente di eseguire una funzione AWS Lambda in risposta a eventi importanti di Amazon Cognito. Amazon Cognito lancia l'evento trigger di sincronizzazione quando viene sincronizzato un set di dati. Puoi utilizzare l'evento del trigger di sincronizzazione per eseguire un'azione quando un utente aggiorna i dati. La funzione è in grado di valutare e, facoltativamente, modificare i dati prima che siano archiviati nel cloud e sincronizzati negli altri dispositivi dell'utente. Questa funzione è utile per convalidare i dati provenienti dal dispositivo prima che siano sincronizzati negli altri dispositivi dell'utente o per aggiornare altri valori nel set di dati in base ai dati in entrata, come l'attribuzione di un riconoscimento quando un giocatore raggiunge un nuovo livello.

Le fasi di seguito ti guideranno attraverso la configurazione di una funzione Lambda che viene eseguita ogni volta che il set di dati di Amazon Cognito viene sincronizzato.

> **Note**
>
> Quando utilizzi gli eventi di Amazon Cognito, puoi usare solo le credenziali ottenute da identità Amazon Cognito. Se hai associato una funzione Lambda ma richiami `UpdateRecords` con credenziali dell'account AWS (credenziali per sviluppatori), la tua funzione Lambda non sarà richiamata.

Creazione di una funzione in AWS Lambda

Per integrare Lambda con Amazon Cognito, per prima cosa hai bisogno di creare una funzione in Lambda. A tale scopo:

Selezione della funzione Lambda in Amazon Cognito

1. Apri la console Lambda.

2. Fai clic su Create a Lambda function (Crea una funzione Lambda).
3. Nella schermata Select blueprint (Seleziona piano), cerca e seleziona "cognito-sync-trigger" ("trigger di sincronizzazione di Cognito");
4. Nella schermata Configure event sources (Configura origini eventi), lascia il tipo di origine impostato su "Cognito Sync Triggering" ("Attivazione della Sincronizzazione di Cognito") e seleziona il tuo pool di identità. Fai clic su Next (Successivo).

> **Note**
>
> Quando si configura un trigger Amazon Cognito Sync all'esterno della console, è necessario aggiungere autorizzazioni basate sulle risorse Lambda per consentire ad Amazon Cognito di richiamare la funzione. È possibile aggiungere questa autorizzazione dalla console Lambda (vedi Utilizzo delle policy basate su risorse per AWS Lambda) o utilizzando l'operazione di Lambda AddPermission.
> Esempio di policy basate su risorse Lambda
> La seguente policy AWS Lambda basata su risorse garantisce ad Amazon Cognito una capacità limitata di richiamare una funzione Lambda. Amazon Cognito può richiamare la funzione solo per conto del pool di identità nella `aws:SourceArn` condizione e dell'account nella condizione `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. Nella schermata Configure function (Configura funzione), inserisci un nome e una descrizione per la funzione. Lascia il Runtime impostato su "Node.js". Lascia il codice invariato per l'esempio. L'esempio predefinito modifica i dati in fase di sincronizzazione. Registra solo il fatto che l'evento trigger di Amazon Cognito Sync si è verificato. Lascia il nome del gestore impostato su "index.handler." Per Ruolo, seleziona un ruolo IAM che concede al tuo codice l'autorizzazione per accedere a AWS Lambda. Per modificare i ruoli, consulta la console IAM. Lascia invariate le impostazioni avanzate. Fai clic su Next (Successivo).
6. Nella schermata Review (Revisione) rivedi i dettagli e fai clic su Create function (Crea funzione). La pagina successiva mostra la tua nuova funzione Lambda.

Ora che disponi di una funzione appropriata scritta in Lambda, devi scegliere quella funzione come gestore dell'evento del trigger di Amazon Cognito Sync. La seguente procedura ti guiderà attraverso il processo.

Dalla home page della console:

1. Fai clic sul nome del pool di identità per cui desideri configurare Amazon Cognito Events. Viene visualizzata la pagina Dashboard (Pannello di controllo) per il tuo pool di identità.
2. Nell'angolo in alto a destra della pagina Dashboard (Pannello di controllo), fai clic su Edit identity pool (Modifica pool di identità). Viene visualizzata la pagina di gestione delle identità federate.
3. Scorri verso il basso e fai clic su Cognito Events (Eventi di Cognito) per espandere l'operazione;
4. Nel menu a discesa del trigger di sincronizzazione, seleziona la funzione Lambda che desideri attivare quando si verifica un evento di sincronizzazione.
5. Fai clic su Save Changes (Salva modifiche).

Ora, la tua funzione Lambda verrà eseguita ogni volta che un set di dati è sincronizzato. La sezione seguente spiega in che modo puoi leggere e modificare i dati nella tua funzione durante la loro fase di sincronizzazione.

Scrittura di una funzione Lambda per trigger di sincronizzazione

I trigger di sincronizzazione seguono il modello di programmazione usato dall'interfaccia del fornitore di servizi. Amazon Cognito fornisce l'input alla tua funzione Lambda nel formato JSON seguente.

```
{
  "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 richiede il valore restituito della funzione nello stesso formato dell'input.

Quando scrivete funzioni per l'evento Sync Trigger, osserva quanto segue:

- Quando Amazon Cognito la tua funzione Lambda durante UpdateRecords, la funzione deve rispondere entro 5 secondi. In caso contrario, il servizio Amazon Cognito Sync genera un'eccezione `LambdaSocketTimeoutException`. Non puoi aumentare il valore di timeout.
- Se ricevi un'eccezione `LambdaThrottledException`, ritenta l'operazione di sincronizzazione per aggiornare i record.
- Amazon Cognito fornisce tutti i record presenti nel set di dati come input per la funzione.
- I record che vengono aggiornati dall'utente dell'app hanno il campo `op` impostato come `replace`. I record eliminati hanno il campo `op` impostato come `remove`.
- Puoi modificare tutti i record, anche se non sono stati aggiornati dall'utente dell'applicazione.
- Tutti i campi, eccetto il datasetRecords, sono di sola lettura. Non modificarli. Se si modificano questi campi, non è possibile aggiornare i record.
- Per modificare il valore di un record, è sufficiente aggiornare il valore e impostare il campo `op` su `replace`.
- Per eliminare un record o impostare il campo `op` su `remove`, imposta un valore nullo.

- Per aggiungere un record, basta aggiungere un nuovo record alla gamma datasetRecords;
- Amazon Cognito ignora qualsiasi record omesso nella risposta quando aggiorna il record.

Funzione Lambda di esempio

Ecco un esempio di funzione Lambda che mostra come accedere, modificare e rimuovere i dati.

```
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);
};
```

Utilizzo della console Amazon Cognito

È possibile utilizzare la console Amazon Cognito per creare e gestire bacini d'utenza e pool di identità.

Questa guida fornisce step-by-step procedure dettagliate per le attività comuni del pool di utenti di Amazon Cognito nella console Amazon Cognito.

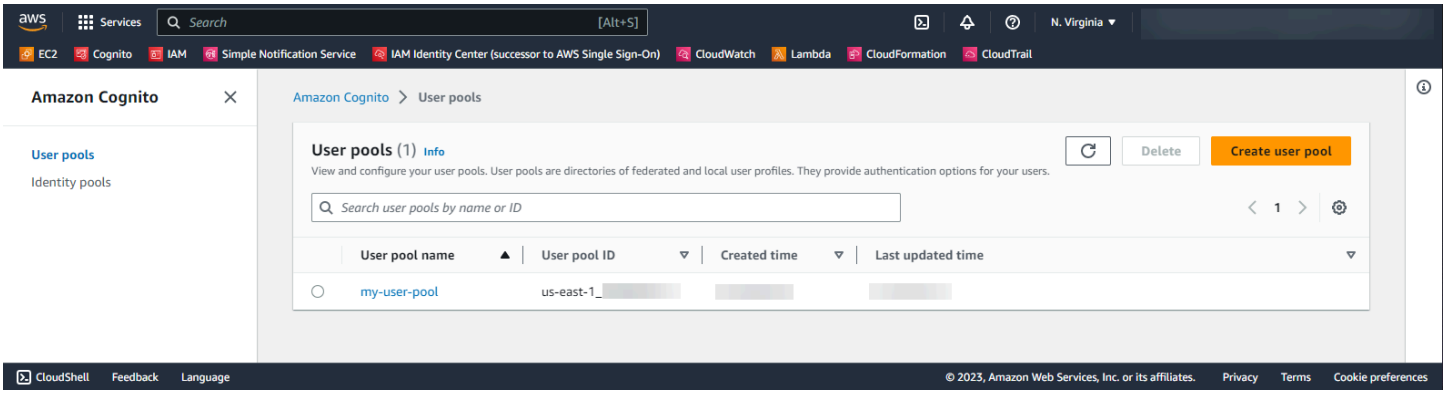
Come utilizzare la console Amazon Cognito

1. Per utilizzare Amazon Cognito, devi registrare un AWS account.
2. Passa alla console Amazon Cognito. È possibile che ti vengano richieste le credenziali AWS .
3. Per creare o modificare un bacino di utenza, scegli Bacini di utenza nel pannello di navigazione a sinistra.

   Per ulteriori informazioni, consulta Nozioni di base sui bacini d'utenza.
4. Per creare o modificare un pool di identità, scegli Pool di identità. Verrai indirizzato alla console originale per i pool di identità Amazon Cognito.

   Per ulteriori informazioni, consulta Guida introduttiva ai pool di identità di Amazon Cognito.

La console Amazon Cognito fa parte di AWS Management Console, che fornisce informazioni sul tuo account e sulla fatturazione. Per ulteriori informazioni, consulta l'argomento relativo all'utilizzo di AWS Management Console.



Argomenti

- La console del pool di utenti
- La console dei pool di identità

La console del pool di utenti

Dalla vista Pool di utenti nella console di Amazon Cognito, scegli un pool di utenti dall'elenco per visualizzare i dettagli. Nella vista dettagliata, la Panoramica del pool di utenti nella parte superiore della console contiene informazioni di base sul pool di utenti. Le seguenti schede consentono di organizzare la configurazione del pool di utenti in funzioni correlate.

Utenti

La scheda Utenti contiene informazioni sugli utenti e sulle importazioni degli utenti da file CSV. Puoi aggiungere, rimuovere e modificare utenti in questa scheda.

Riferimenti

- Gestione degli utenti nel tuo bacino d'utenza
- Importazione di utenti nel bacino d'utenza da un file CSV

Gruppi

La scheda Gruppi contiene informazioni sui gruppi di utenti. Puoi aggiungere, modificare e modificare l'iscrizione ai gruppi e modificare i ruoli IAM associati ai gruppi per l'integrazione del pool di identità.

Riferimenti

- Aggiunta di gruppi a un bacino d'utenza

Esperienza di accesso

La scheda Esperienza di accesso contiene informazioni su come gli utenti accedono al pool di utenti. In questa scheda sono presenti i provider di identità di terze parti, le opzioni per il nome utente, la policy delle password, la configurazione dell'autenticazione a più fattori (MFA), il comportamento di password dimenticata e la memorizzazione del dispositivo. Puoi aggiungere e modificare i provider di identità e modificare il comportamento di accesso complessivo del pool di utenti.

Riferimenti

- Aggiunta di un accesso al bacino d'utenza tramite terze parti
- Personalizzazione degli attributi di accesso
- Aggiunta di requisiti password del bacino d'utenza

- Aggiunta dell'autenticazione MFA a un bacino d'utenza
- Recupero degli account utente
- Utilizzo dei dispositivi utente nel pool di utenti

Esperienza di registrazione

La scheda Esperienza di registrazione contiene informazioni sull'iscrizione self-service, sugli attributi obbligatori, sulla verifica dei numeri di telefono e degli indirizzi e-mail e sugli attributi personalizzati.

Riferimenti

- Registrazione e conferma degli account utente
- Attributi del bacino d'utenza
- Verifica delle informazioni di contatto al momento della registrazione

Messaggistica

La scheda Messaggistica contiene informazioni sui Servizi AWS che desideri utilizzare per inviare messaggi e-mail e SMS agli utenti e sul formato dei messaggi che desideri inviare.

Riferimenti

- Impostazioni e-mail per i bacini d'utenza di Amazon Cognito
- Impostazioni dei messaggi SMS per i bacini d'utenza di Amazon Cognito
- Configurazione dei messaggi di verifica di SMS e-mail e dei messaggi di invito degli utenti

Integrazione di app

La scheda Integrazione di app contiene informazioni sui client dell'app del pool di utenti, sul dominio assegnato agli endpoint del servizio del pool di utenti, sui server di risorse API, sull'interfaccia utente ospitata e sulla sicurezza avanzata. Puoi approfondire ogni client dell'app per configurare quanto segue.

1. Impostazioni dei token
2. URL di callback
3. Flusso di autenticazione
4. Autorizzazioni degli attributi
5. Impostazioni di sicurezza avanzata e dell'interfaccia utente ospitata specifiche dell'app
6. Analisi Amazon Pinpoint

Riferimenti

- Client dell'app pool di utenti
- Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione
- Configurazione di un dominio di bacino d'utenza
- Autorizzazione Scopes, M2M e API con server di risorse
- Aggiunta di sicurezza avanzata a un bacino d'utenza
- Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Proprietà del pool di utenti

La scheda Proprietà del pool di utenti contiene informazioni sulla configurazione del pool di utenti non direttamente correlate agli utenti: trigger Lambda, protezione ACL AWS WAF Web, protezione da eliminazione e tag di risorse.

Riferimenti

- Personalizzazione di flussi di lavoro di bacini d'utenza con trigger Lambda
- Associazione di un ACL Web a un pool di utenti AWS WAF
- Protezione da eliminazione del bacino d'utenza
- Taggare le tue risorse AWS

La console dei pool di identità

Dalla vista Pool di identità nella console di Amazon Cognito, scegli un pool di identità dall'elenco per visualizzare i dettagli. Nella vista dettagliata, la Panoramica del pool di identità nella parte superiore della console contiene informazioni di base sul pool di utenti. Le seguenti schede consentono di organizzare la configurazione del pool di utenti in funzioni correlate.

Statistiche utente

La scheda Statistiche utente mostra informazioni statistiche sugli utenti che hanno generato identità nel pool di identità. Non è possibile configurare alcuna impostazione del pool di identità in questa scheda.

Browser di identità

La scheda Browser di identità contiene informazioni sulle identità individuali generate dagli utenti nel pool di identità. È possibile visualizzare ed eliminare identità.

Riferimenti

- Guida introduttiva ai pool di identità di Amazon Cognito

Accesso utente

La scheda Accesso utente contiene informazioni sui provider di identità collegati al pool di identità, sui provider di sviluppatori, sui ruoli IAM predefiniti assegnati alle identità e sulla configurazione dell'accesso agli ospiti non autenticati. Puoi approfondire ogni provider di identità per configurare quanto segue.

1. Controllo degli accessi basato sul ruolo con Selezione del ruolo IAM
2. Controllo degli accessi basato su attributi con Attributi per il controllo degli accessi

Riferimenti

- Provider di identità esterni con pool di identità
- Ruoli IAM
- Identità autenticate e non autenticate
- Identità autenticate dagli sviluppatori (pool di identità)
- Utilizzo del controllo degli accessi basato su ruoli
- Utilizzo di attributi per il controllo degli accessi

Proprietà del pool di identità

La scheda Proprietà del pool di identità contiene informazioni sulla configurazione di vari pool di identità: autenticazione di base (classica) e tag delle risorse.

- Flusso di autenticazione dei pool di identità (identità federate)
- Taggare le tue risorse AWS

Sicurezza in Amazon Cognito

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili ad Amazon Cognito, consulta AWS Services in Scope by Compliance Program Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon Cognito. Viene illustrato come configurare Amazon Cognito per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Cognito.

Indice

- Protezione dei dati in Amazon Cognito
- Identity and Access Management per Amazon Cognito
- Registrazione e monitoraggio in Amazon Cognito
- Convalida della conformità per Amazon Cognito
- Resilienza in Amazon Cognito
- Sicurezza dell'infrastruttura in Amazon Cognito
- Analisi della configurazione e delle vulnerabilità nei bacini d'utenza di Amazon Cognito
- AWS politiche gestite per Amazon Cognito

Protezione dei dati in Amazon Cognito

Il modello di responsabilità AWS condivisa Modello si applica alla protezione dei dati in Amazon Cognito (Amazon Cognito). Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutto il AWS cloud. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Questo contenuto include le attività di configurazione e gestione della sicurezza per i AWS servizi che utilizzi. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati.

Ai fini della protezione dei dati, ti consigliamo di proteggere le credenziali degli AWS account e di configurare account utente individuali con AWS Identity and Access Management (IAM). In questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il proprio lavoro. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse. AWS
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS dei servizi.
- Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati personali archiviati in Amazon S3.

Ti consigliamo di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, in campi a formato libero, ad esempio un campo Name (Nome). Ciò include quando lavori con Amazon Cognito o altri AWS servizi utilizzando la console, l'API o AWS gli AWS CLI SDK. Gli eventuali dati immessi in Amazon Cognito o altri servizi potrebbero essere prelevati per l'inserimento nei log di diagnostica. Quando fornisci un URL a un server esterno, non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta a tale server.

Crittografia dei dati

La crittografia dei dati in genere rientra in due categorie: crittografia dei dati a riposo e crittografia dei dati in transito.

Crittografia dei dati inattivi

I dati all'interno di Amazon Cognito sono crittografati a riposo in conformità con gli standard del settore.

Crittografia in transito

In quanto servizio gestito, Amazon Cognito è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Cognito attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

I pool di utenti e i pool di identità di Amazon Cognito dispongono di operazioni API autenticate tramite IAM, non autenticate e autorizzate tramite token. Le operazioni API non autenticate e autorizzate tramite token sono destinate all'uso da parte dei clienti, gli utenti finali della tua app. Le operazioni delle API non autenticate o autorizzate da token vengono crittografate a riposo o in transito. Per ulteriori informazioni, consulta Operazioni API autenticate e non autenticate per pool di utenti di Amazon Cognito.

> **Note**
>
> Amazon Cognito crittografa i contenuti internamente e non supporta le chiavi fornite dal cliente.

Identity and Access Management per Amazon Cognito

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (chi può effettuare l'accesso) e autorizzato (chi dispone

delle autorizzazioni) a utilizzare risorse Amazon Cognito. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Funzionamento di Amazon Cognito con IAM
- Esempi di policy basate su identità per Amazon Cognito
- Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito
- Utilizzo di ruoli collegati ai servizi per Amazon Cognito

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon Cognito.

Utente del servizio: se utilizzi il servizio Amazon Cognito per eseguire il lavoro, l'amministratore ti fornirà le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità Amazon Cognito utilizzate per svolgere il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon Cognito, consulta Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito.

Amministratore del servizio: se sei il responsabile delle risorse Amazon Cognito presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon Cognito. Il compito dell'utente è determinare le caratteristiche e le risorse di Amazon Cognito a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon Cognito, consulta Funzionamento di Amazon Cognito con IAM.

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dettagli su come scrivere policy per gestire l'accesso ad Amazon Cognito. Per visualizzare policy basate su identità Amazon Cognito di esempio che puoi utilizzare in IAM, consulta Esempi di policy basate su identità per Amazon Cognito .

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta Cos'è Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente di IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per

effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

- Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruoli IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Funzionamento di Amazon Cognito con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon Cognito, scopri quali funzionalità di IAM sono disponibili per l'uso con Amazon Cognito.

Funzionalità di IAM che puoi utilizzare con Amazon Cognito

Caratteristiche di IAM	Supporto per Amazon Cognito
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	No
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come Amazon Cognito e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Policy basate su identità per Amazon Cognito

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate su identità per Amazon Cognito

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Esempi di policy basate su identità per Amazon Cognito .

Policy basate su risorse all'interno di Amazon Cognito

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account

a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Operazioni delle policy per Amazon Cognito

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di operazioni di Amazon Cognito, consulta Operazioni definite da Amazon Cognito in Riferimento per l'autorizzazione del servizio.

Le operazioni delle policy in Amazon Cognito utilizzano il seguente prefisso prima dell'operazione:

```
cognito-identity
```

Per specificare più operazioni in una sola istruzione, separarle con la virgola.

```
"Action": [
      "cognito-identity:action1",
      "cognito-identity:action2"
         ]
```

API firmate rispetto ad API senza firma

Quando firmi le richieste API di Amazon Cognito con AWS credenziali, puoi limitarle in una policy AWS Identity and Access Management (IAM). Richieste API che è necessario firmare con le credenziali AWS includono l'accesso lato server con `AdminInitiateAuth` e operazioni che creano, visualizzano o modificano le risorse Amazon Cognito come `UpdateUserPool`. Per ulteriori informazioni sulle richieste API firmate, consulta Firmare le richieste AWS API.

Poiché Amazon Cognito è un prodotto di identità dei consumer per le app che si desidera rendere disponibili al pubblico, l'utente ha accesso alle seguenti API non firmate. L'app effettua queste richieste API per gli utenti e i potenziali utenti. Alcune API non richiedono alcuna autorizzazione preventiva, ad esempio, `InitiateAuth` per iniziare una nuova sessione di autenticazione. Alcune API utilizzano token di accesso o chiavi di sessione per l'autorizzazione, come `VerifySoftwareToken` per completare la configurazione MFA per un utente con una sessione autenticata esistente. Un'API dei pool di utenti di Amazon Cognito non firmata e autorizzata supporta una `Session` o il parametro `AccessToken` nella sintassi della richiesta, come mostrato nella Documentazione di riferimento dell'API di Amazon Cognito. Un'API di Amazon Cognito Identity non firmata supporta un parametro `IdentityId` come mostrato nella Documentazione di riferimento dell'API di identità federate di Amazon Cognito.

Per ulteriori informazioni sui modelli di autorizzazione e i ruoli delle operazioni API dei pool di utenti Amazon Cognito, consultare Operazioni API autenticate e non autenticate per pool di utenti di Amazon Cognito.

Operazioni API dei pool di identità di Amazon Cognito

- `GetId`
- `GetOpenIdToken`
- `GetCredentialsForIdentity`
- `UnlinkIdentity`

Operazioni API dei pool di utenti 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`

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Esempi di policy basate su identità per Amazon Cognito .

Risorse di policy per Amazon Cognito

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le operazioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Amazon Resource Names (ARN)

ARN per identità federate di Amazon Cognito

Nei pool di identità (identità federate) di Amazon Cognito è possibile limitare l'accesso di un utente IAM a uno specifico pool di identità attraverso il formato Amazon Resource Name (ARN), come nell'esempio riportato di seguito. Per ulteriori informazioni sugli ARN, consultare Identificatori IAM.

```
arn:aws:cognito-identity:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

ARN per Amazon Cognito Sync

In Amazon Cognito Sync, i clienti possono inoltre limitare l'accesso in base all'ID del pool di identità, all'ID identità e al nome del set di dati.

Per le API che operano su un pool di identità, il formato dell'ARN del pool di identità corrisponde a quello delle identità federate di Amazon Cognito, a eccezione del nome del servizio, ovvero `cognito-sync` al posto di `cognito-identity`:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID
```

Per le API che operano su una identità singola, come `RegisterDevice`, puoi riferirti all'identità individuale con il formato ARN seguente:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID
```

Per le API che operano su set di dati, come `UpdateRecords` e `ListRecords`, puoi riferirti al set di dati individuale utilizzando il formato ARN seguente:

```
arn:aws:cognito-sync:REGION:ACCOUNT_ID:identitypool/IDENTITY_POOL_ID/
identity/IDENTITY_ID/dataset/DATASET_NAME
```

ARN per i bacini d'utenza di Amazon Cognito

Per i pool di utenti di Amazon Cognito, è possibile limitare l'accesso di un utente a un pool di utenti specifico, utilizzando il formato ARN seguente:

```
arn:aws:cognito-idp:REGION:ACCOUNT_ID:userpool/USER_POOL_ID
```

Per visualizzare un elenco di tipi di risorse di Amazon Cognito e i relativi ARN, consulta Risorse definite da Amazon Cognito in Riferimento per l'autorizzazione del servizio. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta Operazioni definite da Amazon Cognito.

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Esempi di policy basate su identità per Amazon Cognito .

Chiavi di condizione delle policy per Amazon Cognito

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se si specificano

più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Amazon Cognito, consulta Chiavi di condizione per Amazon Cognito in Riferimento per l'autorizzazione del servizio. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta Operazioni definite da Amazon Cognito.

Per visualizzare esempi di policy basate su identità Amazon Cognito, consulta Esempi di policy basate su identità per Amazon Cognito .

Liste di controllo degli accessi (ACL) in Amazon Cognito

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato su attributi (ABAC) con Amazon Cognito

Supporta ABAC (tag nelle policy)	Parziale

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è

il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Amazon Cognito

Supporta le credenziali temporanee	Sì

Alcuni Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni del principale tra servizi per Amazon Cognito

Supports forward access sessions (FAS)	No

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per Amazon Cognito

Supporta i ruoli di servizio	Sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

Per ulteriori informazioni sui ruoli di servizio in Amazon Cognito, consulta Attivazione della sincronizzazione push e Sincronizzazione push.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon Cognito. Modifica i ruoli del servizio solo quando Amazon Cognito fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per Amazon Cognito

Supporta i ruoli collegati ai servizi	Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per maggiori dettagli su come creare e gestire i ruoli collegati ai servizi di Amazon Cognito, consulta Utilizzo di ruoli collegati ai servizi per Amazon Cognito.

Esempi di policy basate su identità per Amazon Cognito

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon Cognito. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle operazioni e sui tipi di risorse definiti da Amazon Cognito, incluso il formato degli ARN per ogni tipo di risorse, consulta Operazioni, risorse e chiavi di condizione per Amazon Cognito in Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console Amazon Cognito
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Limitazione dell'accesso alla console ad uno specifico pool di identità
- Consentire l'accesso a un set di dati specifico per tutte le identità di un pool

Best practice per le policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Cognito nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

> **Note**
>
> La versione originale e quella nuova della console Amazon Cognito hanno un comportamento di base diverso quando si visualizzano e si modificano le proprie risorse Amazon Cognito.

Se hai concesso l'autorizzazione alle operazioni nel prefisso del servizio `cognito-idp` solo quando la condizione `aws:ViaAWSService` è True, è possibile che il principale IAM interessato sia stato effettivo per le risorse Amazon Cognito nella console originale, ma non nella nuova console. Per lavorare nella console di Amazon Cognito, non impostare una condizione `aws:ViaAWSService` nelle autorizzazioni di Amazon Cognito nella policy IAM.

Utilizzo della console Amazon Cognito

Per accedere alla console Amazon Cognito, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Cognito presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la console Amazon Cognito, collega anche Amazon `ConsoleAccess` Cognito `ReadOnly` AWS o la policy gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
```

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

Limitazione dell'accesso alla console ad uno specifico pool di identità

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

Consentire l'accesso a un set di dati specifico per tutte le identità di un pool

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

Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Cognito

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon Cognito e IAM.

Argomenti

- Non sono autorizzato a eseguire un'operazione in Amazon Cognito
- Non sono autorizzato a eseguire iam: PassRole
- Sono un amministratore e desidero consentire ad altri utenti di accedere a Amazon Cognito

- Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse di Amazon Cognito

Non sono autorizzato a eseguire un'operazione in Amazon Cognito

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `cognito-identity:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cognito-identity:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `cognito-identity:`*`GetWidget`*.

Se hai bisogno di assistenza, contatta il tuo amministratore. AWS L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se si riceve un errore che indica che non si dispone dell'autorizzazione a eseguire l'operazione `iam:PassRole`, per passare un ruolo a Amazon Cognito è necessario aggiornare le policy.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` prova a utilizzare la console per eseguire un'operazione in Amazon Cognito. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Sono un amministratore e desidero consentire ad altri utenti di accedere a Amazon Cognito

Per consentire ad altri utenti di accedere ad Amazon Cognito, devi creare un'entità IAM (utente o ruolo) per la persona o l'applicazione che ha bisogno di accedere. Tale utente o applicazione utilizzerà le credenziali dell'entità per accedere ad AWS. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in Amazon Cognito.

Per iniziare immediatamente, consulta Creazione dei primi utenti e gruppi delegati IAM nella Guida per l'utente di IAM.

Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse di Amazon Cognito

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Cognito supporta queste funzionalità, consulta Funzionamento di Amazon Cognito con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Utilizzo di ruoli collegati ai servizi per Amazon Cognito

Amazon Cognito utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo unico di ruolo IAM con una politica di fiducia che consente a un utente di assumere il ruolo. Servizio AWS I ruoli collegati ai servizi sono predefiniti da Amazon Cognito e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Amazon Cognito perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon Cognito definisce le autorizzazioni dei ruoli associati ai servizi e, salvo diversamente definito, solo Amazon Cognito può assumere il ruolo. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon Cognito perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruolo collegato ai servizi. Scegli Yes (Sì) in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Amazon Cognito

Amazon Cognito utilizza i seguenti ruoli collegati ai servizi:

- AWSServiceRoleForAmazonCognitoIdpEmailService— Consente al servizio di pool di utenti Amazon Cognito di utilizzare le identità Amazon SES per l'invio di e-mail.

- AWSServiceRoleForAmazonCognitoIdp— Consente ai pool di utenti di Amazon Cognito di pubblicare eventi e configurare endpoint per i tuoi progetti Amazon Pinpoint.

AWSServiceRoleForAmazonCognitoIdpEmailService

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi `AWSServiceRoleForAmazonCognitoIdpEmailService` considera attendibili i seguenti servizi:

- `email.cognito-idp.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Azioni consentite per: AWSServiceRoleForAmazonCognitoIdpEmailService

- Operazione: `ses:SendEmail` e `ses:SendRawEmail`
- Risorsa: *

La policy impedisce a Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Operazioni rifiutate

- Operazione: `ses:List*`
- Risorsa: *

Con queste autorizzazioni, Amazon Cognito può utilizzare gli indirizzi e-mail verificati in Amazon SES solo per inviare e-mail ai tuoi utenti. Amazon Cognito invia e-mail agli utenti quando eseguono alcune operazioni nell'app client per un bacino d'utenza, ad esempio quando accedono o reimpostano una password.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

AWSServiceRoleForAmazonCognitoIdp

Il ruolo AWSServiceRoleForAmazonCognitoIdp collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `email.cognito-idp.amazonaws.com`

La policy delle autorizzazioni del ruolo consente ad Amazon Cognito di completare le seguenti operazioni sulle risorse specificate:

Azioni consentite per AWSServiceRoleForAmazonCognitoIdp

- Operazione: `cognito-idp:Describe`
- Risorsa: *

Con questa autorizzazione, Amazon Cognito può chiamare le operazioni API `Describe` di Amazon Cognito per tuo conto.

> **Note**
>
> Quando integri Amazon Cognito con Amazon Pinpoint utilizzando `createUserPoolClient` e `updateUserPoolClient`, le autorizzazioni della risorsa verranno aggiunte all'SLR come policy in linea. La policy in linea fornirà le autorizzazioni `mobiletargeting:UpdateEndpoint` e `mobiletargeting:PutEvents`. Queste autorizzazioni consentono ad Amazon Cognito di pubblicare eventi e configurare endpoint per i progetti Pinpoint integrati con Cognito.

Creazione di un ruolo collegato ai servizi per Amazon Cognito

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando configuri un pool di utenti per utilizzare la configurazione di Amazon SES per gestire la AWS Management Console consegna delle e-mail nell'API Amazon Cognito AWS CLI, Amazon Cognito crea il ruolo collegato al servizio per te.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando configuri un bacino d'utenza per utilizzare la configurazione di Amazon SES per gestire il recapito e-mail, Amazon Cognito crea il ruolo collegato al servizio per tuo conto.

Prima che Amazon Cognito possa creare questo ruolo, le autorizzazioni IAM utilizzate per configurare il bacino d'utenza devono includere l'operazione `iam:CreateServiceLinkedRole`. Per ulteriori informazioni sull'aggiornamento delle autorizzazioni in IAM, consulta Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente di IAM.

Modifica di un ruolo collegato ai servizi per Amazon Cognito

Non puoi modificare i ruoli AmazonCognitoIdp o AmazonCognitoIdpEmailService quelli collegati al servizio in. AWS Identity and Access Management Dopo aver creato un ruolo collegato al servizio, non puoi modificarne il nome, perché potrebbero farvi riferimento diverse entità. Puoi tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon Cognito

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. Se elimini il ruolo, mantieni solo le entità che Amazon Cognito monitora o mantiene attivamente. Prima di poter eliminare i ruoli AmazonCognitoIdp o AmazonCognitoIdpEmailService quelli collegati al servizio, è necessario eseguire una delle seguenti operazioni per ogni pool di utenti che utilizza il ruolo:

- Eliminare il bacino d'utenza.
- Aggiornare le impostazioni e-mail nel bacino d'utenza in modo da utilizzare la funzionalità e-mail predefinita. L'impostazione di default non utilizza il ruolo collegato al servizio.

Ricordati di eseguire l'azione in ognuno di essi Regione AWS con un pool di utenti che utilizza il ruolo.

> **Note**
>
> Se il servizio Amazon Cognito utilizza tale ruolo quando provi a eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Come eliminare un bacino d'utenza Amazon Cognito

1. Accedi AWS Management Console e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito
2. Scegli Manage User Pools (Gestisci pool di utenti).
3. Nella pagina Your User Pools (I tuoi pool di utenti), scegli il bacino d'utenza da eliminare.
4. Scegli Delete pool (Elimina pool).
5. Nella finestra Delete user pool (Elimina pool di utenti), digita **`delete`** e seleziona Elimina pool.

Per aggiornare un bacino d'utenza di Amazon Cognito in modo che utilizzi la funzionalità e-mail di default

1. Accedi AWS Management Console e apri la console Amazon Cognito all'indirizzo. https://console.aws.amazon.com/cognito

2. Scegli Manage User Pools (Gestisci pool di utenti).
3. Nella pagina Your User Pools (I tuoi pool di utenti), scegli il bacino d'utenza da aggiornare.
4. Nel menu di navigazione a sinistra, seleziona Message customizations (Personalizzazioni di messaggio).
5. Alla voce Do you want to send emails through your Amazon SES Configuration? (Inviare e-mail tramite la configurazione Amazon SES?), seleziona No - Use Cognito (Default) (No, utilizza Cognito (di default)).
6. Una volta completata la configurazione delle opzioni dell'account e-mail, seleziona Save changes (Salva modifiche).

Per eliminare manualmente il ruolo collegato ai servizi utilizzando IAM

Usa la console IAM AWS CLI, o l' AWS API per eliminare ruoli AmazonCognitoIdp o collegati ai AmazonCognitoIdpEmailService servizi. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon Cognito

Amazon Cognito supporta ruoli collegati al servizio Regioni AWS ovunque il servizio sia disponibile. Per ulteriori informazioni, consulta Regioni AWS ed endpoint.

Registrazione e monitoraggio in Amazon Cognito

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Cognito e delle altre AWS soluzioni. Amazon Cognito attualmente supporta i seguenti Servizi AWS in modo da poter monitorare l'organizzazione e le relative attività.

- AWS CloudTrail — Con CloudTrail puoi acquisire chiamate API dalla console Amazon Cognito e da chiamate in codice alle operazioni dell'API Amazon Cognito. Ad esempio, quando un utente si autentica, CloudTrail può registrare dettagli come l'indirizzo IP nella richiesta, chi ha effettuato la richiesta e quando è stata effettuata.
- Amazon CloudWatch Logs: con CloudWatch Logs, puoi inviare log dettagliati delle attività degli utenti a un gruppo di log. Ad esempio, puoi esaminare i log dettagliati delle attività degli utenti per risolvere i problemi relativi alla consegna di e-mail e messaggi SMS agli utenti.
- Amazon CloudWatch Metrics: con le CloudWatch metriche puoi monitorare, segnalare e intraprendere azioni automatiche in caso di evento quasi in tempo reale. Ad esempio, puoi creare CloudWatch dashboard sulle metriche fornite per monitorare i tuoi pool di utenti di Amazon Cognito

oppure puoi CloudWatch creare allarmi sulle metriche fornite per avvisarti in caso di violazione di una soglia prestabilita.

- Amazon CloudWatch Logs Insights: con CloudWatch Logs Insights, puoi configurare l'invio di eventi CloudWatch per CloudTrail il monitoraggio dei file di registro di Amazon CloudTrail Cognito.

Argomenti

- Monitoraggio dei costi
- Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas
- Registrazione delle chiamate all'API Amazon Cognito con AWS CloudTrail

Monitoraggio dei costi

Amazon Cognito addebita i costi per le seguenti dimensioni di utilizzo.

- Pool di utenti: utenti attivi mensili (MAU)
- I MAU del pool di utenti hanno effettuato l'accesso con la federazione OIDC o SAML
- MAU in un pool di utenti con funzionalità di sicurezza avanzate
- Raggruppa utenti attivi tra client di app e volume di richieste di autorizzazione da macchina a macchina (M2M) con assegnazione di credenziali client
- Utilizzo acquistato superiore alle quote predefinite per alcune categorie di API per pool di utenti

Inoltre, le funzionalità del tuo pool di utenti come messaggi e-mail, messaggi SMS e trigger Lambda possono comportare costi in servizi dipendenti. Per una panoramica completa, consulta i prezzi di Amazon Cognito.

Visualizzazione e previsione dei costi

Puoi visualizzare e generare report sui AWS costi nella AWS Billing and Cost Management console. Puoi trovare gli addebiti più recenti per Amazon Cognito nella sezione Fatturazione e pagamenti. In Fatture, addebiti per servizio, filtra `Cognito` per visualizzare l'utilizzo. Per ulteriori informazioni, consulta Visualizzazione della fattura nella Guida per l'utente di AWS Billing .

Per monitorare i tassi di richiesta API, esamina la metrica di utilizzo nella console Service Quotas. Ad esempio, le richieste di credenziali del client vengono visualizzate come Frequenza delle richieste. ClientAuthentication Nella fattura, queste richieste sono associate al client dell'app che le ha

prodotte. Con queste informazioni, è possibile allocare equamente i costi ai tenant in un'architettura multi-tenant.

Per ottenere un conteggio delle richieste M2M per un periodo di tempo, puoi anche inviare AWS CloudTrail eventi a Logs per l'analisi. CloudWatch Interroga i tuoi CloudTrail eventi per verificare la presenza di `Token_POST` eventi con una concessione di credenziali al client. La seguente query CloudWatch Insights restituisce questo conteggio.

```
filter eventName = "Token_POST" and @message like '"grant_type":["client_credentials"]'
 | stats count(*)
```

Gestione dei costi

Le fatture di Amazon Cognito si basano sul numero di utenti, sull'utilizzo delle funzionalità e sul volume delle richieste. Di seguito sono riportati alcuni suggerimenti per gestire i costi in Amazon Cognito,

Non attivare gli utenti inattivi

Le operazioni tipiche che rendono attivo un utente sono l'accesso, la registrazione e la reimpostazione della password. Per un elenco più completo, consulta. Monthly active users (Utenti attivi mensili) Amazon Cognito non conta gli utenti inattivi ai fini della fattura. Evita qualsiasi operazione che renda attivo un utente. Invece dell'operazione AdminGetUserAPI, interroga gli utenti con l'ListUsersoperazione. Non eseguite test amministrativi ad alto volume sulle operazioni del pool di utenti con utenti inattivi.

Collega gli utenti federati

Gli utenti che accedono con un provider di identità SAML 2.0 o OpenID Connect (OIDC) hanno un costo maggiore rispetto agli utenti locali. Puoi collegare questi utenti a un profilo utente locale. Un utente collegato può accedere come utente locale con gli attributi e l'accesso forniti dal proprio utente federato. Gli utenti di SAML o OIDC IdPs che, nel corso di un mese, accedono solo con un account locale collegato vengono fatturati come utenti locali.

Gestisci le tariffe delle richieste

Se il tuo pool di utenti si avvicina al limite massimo della tua quota, potresti prendere in considerazione l'acquisto di capacità aggiuntiva per gestire il volume. Potresti riuscire a ridurre il volume delle richieste nella tua applicazione. Per ulteriori informazioni, consulta Ottimizza le frequenze di richiesta per i limiti di quota.

Richiedi un nuovo token solo quando ne hai bisogno

L'autorizzazione da macchina a macchina (M2M) con concessione di credenziali client può raggiungere un volume elevato di richieste di token. Ogni nuova richiesta di token ha un effetto sulla quota di richiesta e sull'entità della fattura. Per ottimizzare i costi, includi le impostazioni di scadenza dei token e la gestione dei token nella progettazione delle tue applicazioni.

- Memorizza i token di accesso alla cache in modo che, quando l'applicazione richiede un nuovo token, riceva una versione memorizzata nella cache di un token emesso in precedenza. Quando si implementa questo metodo, il proxy di memorizzazione nella cache funge da protezione contro le applicazioni che richiedono token di accesso senza essere consapevoli della scadenza dei token acquisiti in precedenza. I token di caching sono ideali per microservizi di breve durata come le funzioni Lambda e i contenitori Docker.
- Implementa meccanismi di gestione dei token nelle tue applicazioni che tengano conto della scadenza dei token. Non richiedete un nuovo token fino alla scadenza dei token precedenti. Valuta le esigenze di riservatezza e disponibilità di ciascuna applicazione e configura il client dell'app del pool di utenti per emettere token di accesso con un periodo di validità appropriato. La durata dei token personalizzati funziona meglio con API e server più longevi in grado di gestire in modo persistente la frequenza delle richieste di credenziali.

Eliminare le credenziali client non utilizzate (client dell'app)

Le fatture di autorizzazione M2M si basano su due fattori: il tasso di richieste di token e il numero di app client che concedono le credenziali dei clienti. Quando i client di app per l'autorizzazione M2M non sono in uso, eliminali o rimuovi la loro autorizzazione a emettere le credenziali dei client. Per ulteriori informazioni sulla gestione della configurazione del client dell'app, consulta. Client dell'app pool di utenti

Gestire la sicurezza avanzata

Quando configuri funzionalità di sicurezza avanzate in un pool di utenti, la tariffa di fatturazione per la sicurezza avanzata si applica a tutte le MAU del pool di utenti. Se hai utenti che non necessitano di funzionalità di sicurezza avanzate, separali in un altro pool di utenti.

Monitoraggio delle quote e dell'utilizzo in CloudWatch e Service Quotas

Puoi monitorare i pool di utenti di Amazon Cognito utilizzando Amazon CloudWatch o Service Quotas. Puoi anche monitorare l'utilizzo dei pool di identità in Service Quotas. CloudWatch raccoglie

dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. In CloudWatch, puoi impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando tali soglie vengono raggiunte. Per creare un CloudWatch allarme per una quota di servizio, vedi Creare un avviso. CloudWatch I parametri di Amazon Cognito sono disponibili a intervalli di cinque minuti. Per ulteriori informazioni sui periodi di conservazione in CloudWatch, visita la pagina delle CloudWatch domande frequenti di Amazon.

Puoi utilizzare Service Quotas per visualizzare e gestire l'utilizzo delle quote dei pool di utenti e dei pool di identità di Amazon Cognito. La console Service Quotas dispone di tre funzionalità: visualizzazione delle quote di servizio, richiesta di un aumento delle quote di servizio e visualizzazione dell'utilizzo corrente. Puoi utilizzare la prima funzionalità per visualizzare le quote e verificare se la quota è regolabile. Puoi utilizzare la seconda caratteristica per richiedere un aumento delle Service Quotas Infine, puoi utilizzare l'ultima funzionalità per visualizzare l'utilizzo delle quote. Questa funzione è disponibile solo dopo che l'account è rimasto attivo per un po' di tempo. Per ulteriori informazioni sulla visualizzazione delle quote nella console Service Quotas, consulta Visualizzazione di Service Quotas.

> **Note**
>
> I parametri di Amazon Cognito sono disponibili a intervalli di cinque minuti. Per ulteriori informazioni sui periodi di conservazione in CloudWatch, visita la pagina delle CloudWatch domande frequenti di Amazon.

Se hai effettuato l'accesso a un Account AWS account configurato come account di monitoraggio in modalità osservabilità CloudWatch tra account diversi, puoi utilizzare quell'account di monitoraggio per visualizzare le quote di servizio e impostare allarmi per le metriche negli account di origine collegati a quell'account di monitoraggio. Per ulteriori informazioni, consulta osservabilità tra account. CloudWatch

Argomenti

- Registrazione di attività aggiuntive dai pool di utenti di Amazon Cognito
- Parametri per i bacini d'utenza di Amazon Cognito
- Dimensioni per i bacini d'utenza di Amazon Cognito
- Utilizzo della console Service Quotas per tenere traccia dei parametri
- Usa la CloudWatch console per tenere traccia delle metriche
- Crea un CloudWatch allarme per una quota

Registrazione di attività aggiuntive dai pool di utenti di Amazon Cognito

Puoi configurare il tuo pool di utenti per inviare registri dettagliati di alcune attività aggiuntive a un CloudWatch gruppo di registri. Questi registri hanno una granularità più precisa di quelli presenti e possono essere utili per risolvere AWS CloudTrail i problemi del pool di utenti. Quando attivi questa funzionalità, puoi scegliere il gruppo di log a cui Amazon Cognito deve inviare i log. La registrazione dell'attività dell'utente è utile quando desideri scoprire lo stato dei messaggi e-mail e SMS distribuiti dal tuo pool di utenti con Amazon SNS e Amazon SES.

Al momento, puoi fornire log di notifica utente a livello di Errore solo dal tuo pool di utenti.

La registrazione dettagliata non sostituisce né modifica le seguenti funzioni di log dei pool di utenti.

1. CloudTrail registri delle attività di routine degli utenti, come la registrazione e l'accesso.
2. Analisi dell'attività degli utenti su larga scala con metriche. CloudWatch

Separatamente, puoi anche trovare i log dei processi di importazione degli utenti e dei trigger CloudWatch Lambda in Logs. Amazon Cognito e Lambda archiviano questi log in gruppi di log diversi da quelli specificati per i log delle attività dettagliati.

Puoi configurare log di attività dettagliati con l'API dei pool di utenti di Amazon Cognito in SetLogDeliveryConfigurationuna richiesta API. Puoi visualizzare la configurazione di registrazione di un pool di utenti in una GetLogDeliveryConfigurationrichiesta API.

È necessario autorizzare queste richieste con AWS credenziali con le seguenti autorizzazioni.

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

Di seguito è illustrato un evento di esempio da un pool di utenti. Questo schema di log è soggetto a modifiche. Alcuni campi potrebbero essere registrati con valori null.

```
{
    "eventTimestamp": "1687297330677",
    "eventSource": "USER_NOTIFICATION",
    "logLevel": "ERROR",
    "message": {
        "details": "String"
    },
    "logSourceId": {
        "userPoolId": "String"
    }
}
```

La consegna dei log da Amazon Cognito si basa sul miglior tentativo. Il volume di log fornito dal pool di utenti e le quote di servizio per i log possono influire sulla consegna CloudWatch dei log.

CloudWatch I costi dei log si applicano quando la consegna dei log è abilitata. Per ulteriori informazioni, consulta Vending Logs in Amazon CloudWatch Pricing.

Per inviare registri a gruppi di log con una policy delle risorse di dimensione superiore a 5120 caratteri, configura un gruppo di log con un percorso che inizia con `/aws/vendedlogs`. Per ulteriori informazioni, consulta Attivazione della registrazione da determinati servizi. AWS

Parametri per i bacini d'utenza di Amazon Cognito

Nella tabella seguente sono elencati i parametri e le dimensioni disponibili per i bacini d'utenza di Amazon Cognito. Lo spazio dei nomi delle metriche Amazon CloudWatch per Amazon Cognito è `AWS/Cognito`. Per ulteriori informazioni, consulta Namespaces in Amazon CloudWatch User Guide.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando si immette il nome del parametro o i nomi delle dimensioni nella casella di ricerca nella scheda Tutti i parametri nella console. Inoltre, non vengono restituiti nei risultati di un comando list-metrics. Il modo migliore per recuperare queste metriche è con i `get-metric-statistics` comandi `get-metric-data` o nella AWS CLI.

Parametro	Descrizione
`SignUpSuccesses`	Fornisce il numero totale di richieste di registrazione utente riuscite effettuate al bacino d'utenza di Amazon Cognito. Una richiesta di registrazione utente riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Una richiesta con limitazione è considerata anche come una richiesta non riuscita e quindi anche una richiesta con limitazione produrrà un conteggio pari a 0.

Parametro	Descrizione
	Per trovare la percentuale di richieste di registrazione utente riuscite, utilizza la statistic a `Average` di questo parametro. Per calcolare il numero totale di richieste di registrazione utente, utilizza la statistica `Sample Count` su questo parametro. Per calcolare il numero totale di richieste di registrazione utente riuscite, utilizza la statistica `Sum` su questo parametro. Per contare il numero totale di richieste di registrazione utente non riuscite, usa l' CloudWatch `Math`espressione e sottrai la `Sum` statistica dalla statistica. `Sample Count` Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client del bacino d'utenza. Nel caso in cui la registrazione dell'utente venga eseguita da un amministr atore, il parametro viene pubblicato con il client del bacino d'utenza come `Admin`. Si noti che questo parametro non viene emesso per i casi di importazione di utenti e migrazione di utenti. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero

Parametro	Descrizione
`SignUpThrottles`	Fornisce il numero totale di richieste di registrazione utente limitate effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di registrazione utente viene sottoposta a throttling, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di registrazione utente con throttling, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui la richiesta che è stata sottoposta a limitazione è stata effettuata da un amministratore, il parametro viene pubblicato con il client del bacino d'utenza come `Admin`. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero

Parametro	Descrizione
`SignInSuccesses`	Fornisce il numero totale di richieste di autenticazione utente riuscite effettuate al bacino d'utenza di Amazon Cognito. Un'autenticazione utente viene considerata riuscita quando il token di autenticazione viene inviato all'utente. Un'autenticazione riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Una richiesta con limitazione è considerata anche come una richiesta non riuscita e quindi anche una richiesta con limitazione produrrà un conteggio pari a 0. Per trovare la percentuale di richieste di autenticazione utente riuscite, utilizza la statistica `Average` su questo parametro. Per calcolare il numero totale di richieste di autenticazione utente, utilizza la statistica `Sample Count` su questo parametro. Per calcolare il numero totale di richieste di autenticazione utente riuscite, utilizza la statistica `Sum` su questo parametro. Per contare il numero totale di richieste di autenticazione utente non riuscite, usa l' CloudWatch `Math`espressione e sottrai la `Sum` statistica dalla statistica. `Sample Count` Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta.

Parametro	Descrizione
	Si noti che le richieste di aggiornamento del token di Amazon Cognito non sono incluse in questo parametro. Esiste un parametro separato per fornire le statistiche dei token `Refresh`. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero

Parametro	Descrizione
`SignInThrottles`	Fornisce il numero totale di richieste di autenticazione utente limitate effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di autenticazione viene sottoposta a throttling, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di autenticazione utente con throttling, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta. Le richieste di aggiornamento del token Amazon Cognito non sono incluse in questo parametro. Esiste un parametro separato per fornire le statistiche dei token `Refresh`. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero

Parametro	Descrizione
`TokenRefreshSuccesses`	Fornisce il numero totale di richieste riuscite per aggiornare un token Amazon Cognito che sono state effettuate al bacino d'utenza di Amazon Cognito. Una richiesta riuscita di aggiornamento del token Amazon Cognito produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Una richiesta con limitazione è considerata anche come una richiesta non riuscita e quindi anche una richiesta con limitazione produrrà un conteggio pari a 0. Per trovare la percentuale di richieste riuscite di aggiornamento di un token Amazon Cognito, utilizza la statistica `Average` su questo parametro. Per calcolare il numero totale di richieste di aggiornamento di un token Amazon Cognito, utilizza la statistica `Sample Count` su questo parametro. Per calcolare il numero totale di richieste riuscite di aggiornamento di un token Amazon Cognito, utilizza la statistic a `Sum` su questo parametro. Per contare il numero totale di richieste non riuscite di aggiornamento di un token Amazon Cognito, usa l'espressione e CloudWatch `Math` sottrai la statistica `Sum` dalla statistica. `Sample Count` Questo parametro viene pubblicato per ogni client del bacino d'utenza. Se un client del bacino d'utenza non valido è riportato in una richiesta, il valore del client conterrà un valore fisso pari a `Invalid`. Dimensione del parametro: `UserPool`, `UserPoolClient`

Parametro	Descrizione
	Unità: numero
`TokenRefreshThrottles`	Fornisce il numero totale di richieste limitate per aggiornare un token Amazon Cognito che sono state effettuate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di aggiornamento del token Amazon Cognito viene sottoposta a limitazione, viene pubblicato un conteggio di 1 Per calcolare il numero totale di richieste con limitazione di aggiornamento di un token Amazon Cognito, utilizza la statistica `Sum` per questo parametro. Questo parametro viene pubblicato per ogni bacino d'utenza per ogni client. Nel caso in cui con una richiesta venga fornito un client del bacino d'utenza non valido, il valore corrispondente del client del bacino d'utenza nel parametro conterrà un valore `Invalid` fisso anziché il valore non valido effettivo inviato nella richiesta. Dimensione del parametro: `UserPool`, `UserPoolClient` Unità: numero

Parametro	Descrizione
`FederationSuccesses`	Fornisce il numero totale di richieste di federazione delle identità riuscite al bacino d'utenza di Amazon Cognito. Una federazione delle identità è considerata riuscita quando Amazon Cognito emette token di autenticazione per l'utente. Una richiesta di federazione delle identità riuscita produce un valore pari a 1, mentre una richiesta non riuscita produce un valore pari a 0. Le richieste limitate e le richieste che generano un codice di autorizzazione senza token producono un valore pari a 0. Per trovare la percentuale di richieste di federazione delle identità riuscite, utilizza la statistica `Average` di questo parametro. Per calcolare il numero totale di richieste di federazione delle identità, utilizza la statistica `Sample Count` di questo parametro. Per calcolare il numero totale di richieste di federazione delle identità riuscite, utilizza la statistica `Sum` di questo parametro. Per contare il numero totale di richieste di federazione delle identità non riuscite, usa l' CloudWatch `Math`espressione e sottrai la statistica dalla statistica. `Sum` `Sample Count` Dimensione del parametro: `UserPool`, `UserPoolClient` , `IdentityProvider` Unità: numero

Parametro	Descrizione
`FederationThrottles`	Fornisce il numero totale di richieste di federazione delle identità limitate al bacino d'utenza di Amazon Cognito. Ogni volta che una richiesta di federazione delle identità viene limitata, viene pubblicato un conteggio di 1. Per calcolare il numero totale di richieste di federazione delle identità con throttling, utilizza la statistica `Sum` per questo parametro. Dimensione del parametro: `UserPool`, `UserPoolClient` , `IdentityProvider` Unità: numero
`CallCount`	Fornisce il numero totale di chiamate effettuate dai clienti in relazione a una categoria. Questo parametro include tutte le chiamate, ad esempio chiamate limitate, chiamate non riuscite e chiamate riuscite. Questo parametro è disponibile nello `nameSpace` Usage (Utilizzo). La quota di categoria viene applicata per ogni AWS account in tutti i pool di utenti di un account e di una regione. Per contare il numero totale di chiamate in una categoria puoi utilizzare la statistica `Sum` per questo parametro. Dimensione parametro: Servizio, Tipo, Risorsa, Classe Unità: numero

Parametro	Descrizione
`ThrottleCount`	Fornisce il numero totale di chiamate limitate in relazione a una categoria. Questo parametro è disponibile nello `nameSpace` Usage (Utilizzo). Questo parametro viene pubblicato a livello di account. Per contare il numero totale di chiamate in una categoria puoi utilizzare la statistica `Sum` per questo parametro. Dimensione parametro: Servizio, Tipo, Risorsa, Classe Unità: numero

Dimensioni per i bacini d'utenza di Amazon Cognito

Le seguenti dimensioni vengono utilizzate per perfezionare i parametri di utilizzo pubblicati da Amazon Cognito. Le dimensioni si applicano solo ai parametri `CallCount` e `ThrottleCount` .

Dimensione	Descrizione
Servizio	Il nome del AWS servizio che contiene la risorsa. Per i parametri di utilizzo di Amazon Cognito, il valore per questa dimensione è `Cognito user pool`.
Type	Il tipo di entità che viene segnalato. Attualmen te, l'unico valore valido per i parametri di utilizzo di Amazon Cognito è API.
Risorsa	Il tipo di risorsa in esecuzione. L'unico valore valido è il nome della categoria.

Dimensione	Descrizione
Classe	La classe della risorsa monitorata. Amazon Cognito non utilizza la dimensione della classe.

Utilizzo della console Service Quotas per tenere traccia dei parametri

Puoi visualizzare e gestire le quote dei pool di utenti e dei pool di identità di Amazon Cognito da una posizione centrale con Service Quotas. Puoi utilizzare la console Service Quotas per visualizzare i dettagli su una quota specifica, monitorare l'utilizzo della quota e richiedere un aumento della quota. Per alcuni tipi di quote, puoi creare un CloudWatch allarme per monitorare l'utilizzo delle quote. Per ulteriori informazioni sulle metriche di Amazon Cognito che puoi monitorare, consultare Monitoraggio dell'uso delle quote.

Per visualizzare l'utilizzo di Service Quotas dei pool di utenti e dei pool di identità di Amazon Cognito, completa i seguenti passaggi.

1. Apri la console Service Quotas.
2. Nel pannello di navigazione, scegli Servizi AWS (servizi AWS).
3. Dall'elenco dei Servizi AWS , cerca e scegli i pool di utenti di Amazon Cognito o le identità federate di Amazon Cognito. Viene visualizzata la pagina Quote di servizio.
4. Seleziona una quota che supporti il CloudWatch monitoraggio. Ad esempio, scegli `Rate of UserAuthentication requests` nei pool di utenti di Amazon Cognito.
5. Scorri verso il basso fino a Monitoring (Monitoraggio). Questa sezione viene visualizzata solo per le quote che supportano il CloudWatch monitoraggio.
6. In Monitoring (Monitoraggio) puoi visualizzare l'utilizzo corrente delle quote di servizio nel grafico.
7. In Monitoring (Monitoraggio) seleziona un'ora, tre ore, dodici ore, un giorno, tre giorni o una settimana.
8. Seleziona una qualsiasi area all'interno del grafico per visualizzare la percentuale di utilizzo delle quote di servizio. Da qui, puoi aggiungere il grafico alla dashboard o utilizzare il menu delle azioni per selezionare Visualizza nelle metriche, che ti porterà alle metriche correlate nella console. CloudWatch

Usa la CloudWatch console per tenere traccia delle metriche

Puoi tracciare e raccogliere i parametri dei pool di utenti di Amazon Cognito utilizzando. CloudWatch La CloudWatch dashboard mostrerà le metriche relative a ogni AWS servizio che utilizzi. Puoi usarlo CloudWatch per creare allarmi metrici. Gli allarmi possono essere configurati per inviare notifiche all'utente o apportare una modifica a una risorsa specifica che si sta monitorando. Per visualizzare le metriche delle quote di servizio in CloudWatch, completa i seguenti passaggi.

1. Apri la CloudWatch console.
2. Nel pannello di navigazione, seleziona Metrics (Parametri).
3. In All metrics (Tutti i parametri) seleziona un parametro e una dimensione.
4. Seleziona la casella di controllo accanto a un parametro. I parametri saranno mostrati nel grafico.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando digiti il nome del parametro o i nomi delle dimensioni nella casella di ricerca della scheda Tutti i parametri della console e non vengono restituiti nei risultati del comando list-metrics. Il modo migliore per recuperare questi parametri è con i comandi `get-metric-data` o `get-metric-statistics` in AWS CLI.

Crea un CloudWatch allarme per una quota

Amazon Cognito fornisce metriche di CloudWatch utilizzo che corrispondono alle quote di AWS servizio e alle API. `CallCount` `ThrottleCount` Per ulteriori informazioni sul monitoraggio dell'utilizzo in, consulta. CloudWatch Monitoraggio dell'uso delle quote

Nella console Service Quotas, puoi configurare gli allarmi che avvisano quando il tuo utilizzo si avvicina a una quota di servizio. Per informazioni su come configurare un CloudWatch allarme utilizzando la console Service Quotas, vedi Service Quotas and alarms. CloudWatch

Registrazione delle chiamate all'API Amazon Cognito con AWS CloudTrail

Amazon Cognito è integrato con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Cognito. CloudTrail acquisisce un

sottoinsieme di chiamate API per Amazon Cognito come eventi, incluse le chiamate dalla console Amazon Cognito e le chiamate di codice alle operazioni dell'API Amazon Cognito. Se crei un trail, puoi scegliere di distribuire CloudTrail eventi a un bucket Amazon S3, inclusi eventi per Amazon Cognito. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon Cognito, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e attivarlo, consulta la Guida per l'AWS CloudTrail utente.

Puoi anche creare CloudWatch allarmi Amazon per CloudTrail eventi specifici. Ad esempio, puoi impostare l'attivazione CloudWatch di un allarme se viene modificata la configurazione di un pool di identità. Per ulteriori informazioni, consulta Creazione di CloudWatch allarmi per CloudTrail eventi: esempi.

Argomenti

- Informazioni su Amazon Cognito in CloudTrail
- Informazioni sugli eventi di accesso di Amazon Cognito
- Analisi degli eventi di Amazon CloudTrail Cognito con Amazon CloudWatch Logs Insights

Informazioni su Amazon Cognito in CloudTrail

CloudTrail si attiva quando crei il tuo Account AWS. Quando si verifica un'attività di evento supportata in Amazon Cognito, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Cognito, crea un percorso. Un CloudTrail trail invia i file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni . Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati

- Configurazione delle notifiche di Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'elemento CloudTrail userIdentity.

Dati riservati in AWS CloudTrail

Poiché i pool di utenti e i pool di identità elaborano i dati degli utenti, Amazon Cognito oscura alcuni campi privati dei tuoi CloudTrail eventi con il valore. `HIDDEN_FOR_SECURITY_REASONS` Per esempi di campi che non vengono compilati da Amazon Cognito negli eventi, consulta Informazioni sugli eventi di accesso di Amazon Cognito. Amazon Cognito oscura solo alcuni campi che contengono di solito informazioni utente, come password e token. Amazon Cognito non esegue alcun rilevamento o mascheratura automatica di informazioni identificative personali inserite in campi non privati nelle richieste API.

Bacini d'utenza di Amazon Cognito

Amazon Cognito supporta la registrazione di tutte le azioni elencate nella pagina delle azioni del pool di utenti come eventi nei CloudTrail file di registro. Amazon Cognito registra gli eventi del pool di utenti CloudTrail come eventi di gestione.

Il `eventType` campo in una CloudTrail voce relativa ai pool di utenti di Amazon Cognito indica se l'app ha effettuato la richiesta all'API dei pool di utenti di Amazon Cognito o a un endpoint che fornisce risorse per OpenID Connect, SAML 2.0 o l'interfaccia utente ospitata. Le richieste API hanno un `eventType` di `AwsApiCall` e le richieste degli endpoint hanno un `eventType` di `AwsServiceEvent`.

Amazon Cognito registra le seguenti richieste di interfaccia utente ospitata nell'interfaccia utente ospitata come eventi in. CloudTrail

Operazioni dell'interfaccia utente ospitate in CloudTrail

Operazione	Descrizione
`Login_GET` , `CognitoAuthentication`	Un utente visualizza o invia le credenziali al tuo Endpoint Login.
`OAuth2_Authorize_GET` , `Beta_Authorize_GET`	Un utente visualizza il tuo Endpoint Authorize.
`OAuth2Response_GET` , `OAuth2Response_POST`	Un utente invia un token IdP al tuo endpoint `/oauth2/idpresponse` .
`SAML2Response_POST` , `Beta_SAML2Response_POST`	Un utente invia un'asserzione SAML IdP al tuo endpoint `/saml2/idpresponse` .
`Login_OIDC_SAML_POST`	Un utente specifica un nome utente nel tuo Endpoint Login e lo abbina con un identificatore IdP.
`Token_POST` , `Beta_Token_POST`	Un utente invia un codice di autorizzazione al tuo Endpoint Token.
`Signup_GET` , `Signup_POST`	Un utente invia le informazioni di registrazione al tuo endpoint `/signup`.
`Confirm_GET` , `Confirm_POST`	Un utente invia un codice di conferma nell'interfaccia utente ospitata.
`ResendCode_POST`	Un utente invia una richiesta per inviare nuovamente un codice di conferma nell'interfaccia utente ospitata.
`ForgotPassword_GET` , `ForgotPassword_POST`	Un utente invia una richiesta per reimpostare la relativa password sul tuo endpoint `/forgotPassword` .
`ConfirmForgotPassword_GET` , `ConfirmForgotPassword_POST`	Un utente invia un codice al tuo endpoint `/confirmForgotPassword` che conferma la sua richiesta `ForgotPassword` .

Operazione	Descrizione
`ResetPassword_GET` , `ResetPassword_POST`	Un utente invia una nuova password nell'interfaccia utente ospitata.
`Mfa_GET`, `Mfa_POST`	Un utente invia un codice di autenticazione a più fattori (MFA) nell'interfaccia utente ospitata.
`MfaOption_GET` , `MfaOption_POST`	Un utente sceglie il metodo preferito per l'MFA nell'interfaccia utente ospitata.
`MfaRegister_GET` , `MfaRegister_POST`	Un utente invia un codice di autenticazione a più fattori (MFA) nell'interfaccia utente ospitata durante la registrazione dell'MFA.
`Logout`	Un utente si disconnette dal tuo endpoint `/logout`.
`SAML2Logout_POST`	Un utente si disconnette dal tuo endpoint `/saml2/logout` .
`Error_GET`	Un utente visualizza una pagina di errore nell'interfaccia utente ospitata.
`UserInfo_GET` , `UserInfo_POST`	Un utente o un IdP scambia informazioni con il tuo Endpoint UserInfo.
`Confirm_With_Link_GET`	Un utente invia una conferma in base a un link inviato da Amazon Cognito in un messaggio e-mail.
`Event_Feedback_GET`	Un utente invia un feedback ad Amazon Cognito in merito a un evento delle funzionalità di sicurezza avanzata.

> **Note**
>
> Amazon Cognito registra`UserSub`, ma non `UserName` nei CloudTrail log, le richieste specifiche di un utente. È possibile trovare un utente per un dato `UserSub`richiamando l'API `ListUsers` e utilizzando un filtro per sub.

Pool di identità di Amazon Cognito

Eventi di dati

Amazon Cognito registra i seguenti eventi di Amazon Cognito Identity come eventi di dati. CloudTrail Gli eventi relativi ai dati sono operazioni API data-plane ad alto volume che CloudTrail non vengono registrate per impostazione predefinita. Per gli eventi di dati sono previsti costi aggiuntivi.

- GetCredentialsForIdentity
- GetId
- GetOpenIdToken
- GetOpenIdTokenForDeveloperIdentity
- UnlinkIdentity

Per generare CloudTrail log per queste operazioni API, devi attivare gli eventi relativi ai dati nel tuo percorso e scegliere i selettori di eventi per i pool di identità di Cognito. Per ulteriori informazioni, consulta Registrazione di eventi di dati per i percorsi nella Guida per l'utente di AWS CloudTrail .

Puoi anche aggiungere i selettori di eventi dei pool di identità al tuo trail con il seguente comando CLI.

```
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\"\
         ]\
      }\
   ]\
}"
```

Eventi di gestione

Amazon Cognito registra il resto delle operazioni API dei pool di identità di Amazon Cognito come eventi di gestione. CloudTrail registra le operazioni API degli eventi di gestione per impostazione predefinita.

Per un elenco delle operazioni delle API dei pool di identità di Amazon Cognito a cui Amazon Cognito accede, consulta il riferimento CloudTrail all'API dei pool di identità di Amazon Cognito.

Amazon Cognito Sync

Amazon Cognito registra tutte le operazioni API di Amazon Cognito Sync come eventi di gestione. Per un elenco delle operazioni dell'API Amazon Cognito Sync a cui Amazon Cognito accede, consulta il riferimento CloudTrail all'API Amazon Cognito Sync.

Informazioni sugli eventi di accesso di Amazon Cognito

Un trail può fornire eventi come file di log a un bucket Amazon S3 da te specificato. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Argomenti

- CloudTrail Eventi di esempio per una registrazione all'interfaccia utente ospitata
- CloudTrail Evento di esempio per una richiesta SAML
- CloudTrail Eventi di esempio per le richieste all'endpoint del token
- CloudTrail Evento di esempio per CreateIdentityPool
- CloudTrail Evento di esempio per GetCredentialsForIdentity
- CloudTrail Evento di esempio per GetId
- CloudTrail Evento di esempio per GetOpenIdToken

- CloudTrail Evento di esempio per GetOpenIdTokenForDeveloperIdentity
- CloudTrail Evento di esempio per UnlinkIdentity

CloudTrail Eventi di esempio per una registrazione all'interfaccia utente ospitata

CloudTrail Gli eventi di esempio seguenti mostrano le informazioni che Amazon Cognito registra quando un utente si registra tramite l'interfaccia utente ospitata.

Amazon Cognito registra il seguente evento quando un nuovo utente accede alla pagina di accesso della tua app.


```
{
    "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 registra il seguente evento quando un nuovo utente sceglie Sign up (Registrazione) dalla pagina di accesso della tua app.

```
{
    "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 registra il seguente evento quando un nuovo utente sceglie un nome utente, immette un indirizzo e-mail e sceglie una password dalla pagina di accesso dell'app. Amazon Cognito non registra le informazioni di identificazione sull'identità dell'utente. 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 registra il seguente evento quando un nuovo utente accede alla pagina di conferma dell'utente nell'interfaccia utente ospitata dopo la registrazione.

```
{
    "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_aaaaaaaaa"
    },
    "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 registra il seguente evento quando, nella pagina di conferma dell'utente nell'interfaccia utente ospitata, un utente immette un codice inviato da Amazon Cognito in un messaggio 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_aaaaaaaaa"
    },
    "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"
}
```

CloudTrail Evento di esempio per una richiesta SAML

Amazon Cognito registra il seguente evento quando un utente che si è autenticato con il tuo IdP SAML invia l'asserzione SAML al tuo endpoint `/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"
}
```

CloudTrail Eventi di esempio per le richieste all'endpoint del token

Di seguito sono riportati eventi di esempio dalle richieste all'Endpoint Token.

Amazon Cognito registra il seguente evento quando un utente che ha autenticato e ricevuto un codice di autorizzazione invia il codice al tuo endpoint `/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 registra il seguente evento quando il sistema di back-end invia una richiesta `client_credentials` per un token di accesso all'endpoint `/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-a5b3-4578-9ad6-3dfdcb8a6d34",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "serviceEventDetails":
    {
        "serviceAccountId": "111122223333"
    },
    "eventCategory": "Management"
}
```

Amazon Cognito registra il seguente evento quando la tua app scambia un token di aggiornamento con un nuovo ID e un token di accesso con il tuo endpoint `/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"
}
```

CloudTrail Evento di esempio per CreateIdentityPool

Il seguente esempio è una voce di log per una richiesta dell'operazione `CreateIdentityPool`. La richiesta è stata effettuata da un utente IAM denominato 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": "000000000000000"
        }
    },
    "responseElements": {
        "identityPoolName": "TestPool",
        "identityPoolId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE",
        "allowUnauthenticatedIdentities": true,
        "supportedLoginProviders": {
            "graph.facebook.com": "000000000000000"
        }
    },
    "requestID": "15cc73a1-0780-460c-91e8-e12ef034e116",
    "eventID": "f1d47f93-c708-495b-bff1-cb935a6064b2",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

CloudTrail Evento di esempio per GetCredentialsForIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `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": {
        "credentials": {
            "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
            "sessionToken": "aAaAaAaAaAaAab1111111111EXAMPLE",
            "expiration": "Jan 19, 2023 5:55:08 PM"
        },
        "identityId": "us-east-1:1cf667a2-49a6-454b-9e45-23199EXAMPLE"
    },
    "requestID": "659dfc23-7c4e-4e7c-858a-1abce884d645",
    "eventID": "6ad1c766-5a41-4b28-b5ca-e223ccb00f0d",
    "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"
}
```

CloudTrail Evento di esempio per GetId

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetId`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-01-19T16:55:05Z",
    "eventSource": "cognito-identity.amazonaws.com",
    "eventName": "GetId",
    "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-id",
```

```
    "requestParameters": {
        "identityPoolId": "us-east-1:2dg778b3-50b7-565c-0f56-34200EXAMPLE",
        "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": "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"
}
```

CloudTrail Evento di esempio per GetOpenIdToken

Il seguente esempio è una voce di log per una richiesta dell'operazione `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"
}
```

CloudTrail Evento di esempio per GetOpenIdTokenForDeveloperIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `GetOpenIdTokenForDeveloperIdentity`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA1EXAMPLE:johns-AssumedRoleSession",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/johns-AssumedRoleSession",
        "accountId": "111122223333",
        "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA1EXAMPLE",
                "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"
}
```

CloudTrail Evento di esempio per UnlinkIdentity

Il seguente esempio è una voce di log per una richiesta dell'operazione `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_aaaaaaaaa":
 "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_aaaaaaaaa"]
    },
    "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"
}
```


Analisi degli eventi di Amazon CloudTrail Cognito con Amazon CloudWatch Logs Insights

Puoi cercare e analizzare i tuoi CloudTrail eventi di Amazon Cognito con Amazon CloudWatch Logs Insights. Quando configuri il percorso per inviare eventi a CloudWatch Logs, CloudTrail invia solo gli eventi che corrispondono alle impostazioni del percorso.

Per interrogare o ricercare CloudTrail gli eventi di Amazon Cognito, nella CloudTrail console, assicurati di selezionare l'opzione Eventi di gestione nelle impostazioni del percorso in modo da poter

monitorare le operazioni di gestione eseguite sulle tue AWS risorse. Se lo desideri, puoi selezionare l'opzione Eventi Insights nelle impostazioni del percorso per identificare errori, attività insolite o comportamenti insoliti dell'utente nell'account.

Query di Amazon Cognito di esempio

Puoi utilizzare le seguenti query nella CloudWatch console Amazon.

Query generali

Trova i 25 log eventi aggiunti più di recente.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com"
```

Ottieni un elenco dei 25 eventi di log aggiunti più di recente che includono eccezioni.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and @message like /Exception/
```

Query di eccezioni ed errori

Trova i 25 eventi di log aggiunti più di recente con codice di errore `NotAuthorizedException` insieme al bacino d'utenza di Amazon Cognito `sub`.

```
fields @timestamp, additionalEventData.sub as user | sort @timestamp desc | limit 25
| filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
```

Trova il numero di record con `sourceIPAddress` e `eventName` corrispondente.

```
filter eventSource = "cognito-idp.amazonaws.com"
| stats count(*) by sourceIPAddress, eventName
```

Trova i primi 25 indirizzi IP che hanno attivato un errore `NotAuthorizedException`.

```
filter eventSource = "cognito-idp.amazonaws.com" and errorCode=
 "NotAuthorizedException"
| stats count(*) as count by sourceIPAddress, eventName
```

```
| sort count desc | limit 25
```

Trova i primi 25 indirizzi IP che hanno chiamato l'API `ForgotPassword`.

```
filter eventSource = "cognito-idp.amazonaws.com" and eventName = 'ForgotPassword'
| stats count(*) as count by sourceIPAddress
| sort count desc | limit 25
```

Convalida della conformità per Amazon Cognito

I revisori di terze parti valutano la sicurezza e la conformità di Amazon Cognito nell'ambito di AWS diversi programmi di conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta i AWS servizi rientranti nell'ambito dei programmi di conformità, . Per informazioni generali, consulta Programmi per la conformità di AWS.

È possibile scaricare i report di controllo di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La responsabilità di conformità durante l'utilizzo di Amazon Cognito è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWS mette a disposizione le seguenti risorse:

- Guide Quick Start per la sicurezza e la conformità. Queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive in che modo le aziende possono utilizzare per creare applicazioni conformi all'HIPAA. AWS
- AWS risorse per la conformità e risorse per la conformità: questa raccolta di riguardare il settore e la località in cui operate.
- Valutazione delle risorse in base alle regole contenute nella Guida per gli AWS Config sviluppatori: AWS Config valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza in Amazon Cognito

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta infrastruttura AWS globale.

Argomenti

- Considerazioni sui dati regionali

Considerazioni sui dati regionali

I pool di utenti di Amazon Cognito vengono creati ciascuno in una AWS regione e archiviano i dati del profilo utente solo in quella regione. I pool di utenti possono inviare dati utente a una AWS regione diversa, a seconda di come sono configurate le funzionalità opzionali.

- Se l'impostazione di default `no-reply@verificationemail.com` di indirizzo e-mail viene utilizzata per la verifica di instradamento degli indirizzi e-mail dei bacini d'utenza di Amazon Cognito, le e-mail vengono instradate nella stessa regione del bacino d'utenza associato.
- Se viene utilizzato un indirizzo e-mail diverso per configurare Amazon Simple Email Service (Amazon SES) con i pool di utenti di Amazon Cognito, tale indirizzo e-mail viene instradato AWS attraverso la regione associata all'indirizzo e-mail in Amazon SES.
- I messaggi SMS provenienti dai bacini d'utenza di Amazon Cognito vengono instradati attraverso la stessa regione Amazon SNS, a meno che non sia diversamente indicato in Configurazione della verifica di e-mail o telefono.
- Se si utilizza l'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito, i dati di eventi vengono instradati alla regione Stati Uniti orientali (Virginia settentrionale).

> **Note**
>
> Amazon Pinpoint è disponibile in diverse AWS regioni in Nord America, Europa, Asia e Oceania. Le regioni di Amazon Pinpoint includono l'API Amazon Pinpoint. Se Amazon

Pinpoint è supportato da Amazon Cognito, allora Amazon Cognito invierà eventi ai progetti Amazon Pinpoint all'interno della stessa regione Amazon Pinpoint. Se una regione non è supportata da Amazon Pinpoint, Amazon Cognito supporterà solo l'invio di eventi in us-east-1. Per informazioni dettagliate sulla regione di Amazon Pinpoint, consultare endpoint e quote di Amazon Pinpoint e Utilizzo delle analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito.

Sicurezza dell'infrastruttura in Amazon Cognito

In quanto servizio gestito, Amazon Cognito è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Cognito attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Analisi della configurazione e delle vulnerabilità nei bacini d'utenza di Amazon Cognito

AWS gestisce attività di sicurezza di base come l'applicazione di patch al sistema operativo (OS) guest e al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta le seguenti risorse :

- Convalida della conformità per Amazon Cognito
- Modello di responsabilità condivisa

AWS politiche gestite per Amazon Cognito

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite che scrivere le politiche da soli. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

Per fornire l'accesso ad Amazon Cognito sono disponibili diverse policy tramite la console IAM:

- `AmazonCognitoPowerUser`: autorizzazioni per accedere e gestire tutti gli aspetti dei tuoi pool di identità e bacini d'utenza. Per visualizzare le autorizzazioni relative a questa politica, vedere. AmazonCognitoPowerUser
- `AmazonCognitoReadOnly`: autorizzazioni per accesso in sola lettura ai tuoi pool di identità e ai tuoi bacini d'utenza. Per visualizzare le autorizzazioni relative a questa politica, vedere. AmazonCognitoReadOnly
- `AmazonCognitoDeveloperAuthenticatedIdentities`: autorizzazioni per il tuo sistema di autenticazione a integrarsi con Amazon Cognito. Per visualizzare le autorizzazioni relative a questa politica, vedere. AmazonCognitoDeveloperAuthenticatedIdentities

Queste policy sono gestite dal team di Amazon Cognito, per cui gli utenti continueranno ad avere lo stesso livello di accesso anche se verranno aggiunte nuove API.

> **Note**
>
> Quando crei un nuovo pool di identità, puoi creare automaticamente nuovi ruoli per l'accesso utente utenti autenticato e guest. L'amministratore che crea il pool di identità con nuovi ruoli IAM deve anche disporre delle autorizzazioni IAM per creare i ruoli.

I pool di identità con accesso guest non autenticato applicano una policy AWS gestita aggiuntiva`AmazonCognitoUnAuthedIdentitiesSessionPolicy`, come policy di sessione agli utenti non autenticati. Questa politica AWS gestita non ha alcun uso amministrativo previsto. Limita invece l'ambito delle autorizzazioni che è possibile applicare agli utenti guest nel flusso di autenticazione migliorato del pool di identità. Per ulteriori informazioni, consulta Ruoli IAM.

Amazon Cognito si aggiorna alle AWS politiche gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon Cognito da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivere il feed RSS nella pagina Document history (Cronologia dei documenti) di Amazon Cognito.

Modifica	Descrizione	Data
`AmazonCognitoUnAuthedIdentitiesSessionPolicy` : nuova policy	È stata aggiunta una policy AWS gestita per la riduzione dei privilegi degli utenti ospiti nei pool di identità.	14 luglio 2023
`AmazonCognitoPowerUser` e `AmazonCog`	Sono state aggiunte nuove autorizzazioni per consentire agli utenti esperti di visualizz	19 luglio 2022

Modifica	Descrizione	Data
`nitoReadOnly` : aggiornamenti alle policy esistenti	are e gestire le associazioni di ACL AWS WAF Web ai pool di utenti di Amazon Cognito. Sono state aggiunte nuove autorizzazioni per consentire agli utenti di sola lettura di visualizzare le associazioni degli ACL AWS WAF Web ai pool di utenti di Amazon Cognito.	
`AmazonCognitoPowerUser` : aggiornamento a una policy esistente	Aggiunta una nuova autorizzazione per consentire ad Amazon Cognito di chiamare le operazioni `PutIdentityPolicy` e `ListConfigurationSets` di Amazon Simple Notification Service. Questa modifica consente ai bacini d'utenza di Amazon Cognito di aggiornare le policy di autorizzazione di invio di Amazon SES e di applicare i set di configurazione Amazon SES quando configuri l'invio di e-mail nel bacino d'utenza.	17 novembre 2021

Modifica	Descrizione	Data
`AmazonCognitoPowerUser` : aggiornamento a una policy esistente	Aggiunta una nuova autorizzazione per consentire ad Amazon Cognito di chiamare l'operazione `GetSMSSandboxAccountStatus` di Amazon Simple Notification Service. Questa modifica consente ai bacini d'utenza di Amazon Cognito di decidere se è necessario uscire dalla sandbox di Amazon Simple Notification Service per inviare messaggi a tutti gli utenti finali tramite i bacini d'utenza.	1° giugno 2021
Amazon Cognito ha iniziato a monitorare le modifiche	Amazon Cognito ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	1 marzo 2021

Assegnazione di tag alle risorse Amazon Cognito

Un tag è un'etichetta di metadati che l'utente o AWS assegna a una risorsa AWS. Ciascun tag è formato da una chiave e da un valore, Per i tag assegnati da te, puoi definire la chiave e il valore. Ad esempio, potresti definire la chiave come `stage` e il valore di una risorsa come `test`.

I tag consentono di eseguire le seguenti operazioni:

- Identificare e organizzare le risorse AWS. Molti servizi AWS supportano l'assegnazione di tag, perciò è possibile assegnare lo stesso tag a risorse di diversi servizi. Questo ti aiuta a indicare quali risorse sono correlate. Ad esempio, puoi assegnare a un bacino d'utenza Amazon Cognito lo stesso tag che si assegna a una tabella Amazon DynamoDB.
- Tenere traccia dei costi AWS. I tag vengono attivati nel pannello di controllo AWS Billing and Cost Management. AWS usa i tag per l'allocazione dei costi per categorizzare i costi e fornire un report di allocazione dei costi mensili. Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella AWS BillingGuida per l'utente.
- Controllare gli accessi alle risorse in base ai tag a loro assegnati. Puoi controllare gli accessi specificando chiavi e valori di tag nelle condizioni di una policy AWS Identity and Access Management (IAM). Ad esempio, puoi consentire a un utente di aggiornare un pool di utenti, ma solo se questo dispone di un tag `owner` con un valore del nome di tale utente. Per ulteriori informazioni, consulta Controllo degli accessi tramite tag nella Guida per l'utente di IAM.

Puoi utilizzare la AWS Command Line Interface o l'API di Amazon Cognito per aggiungere, modificare o eliminare tag per i bacini d'utenza e i pool di identità. In particolare per i bacini d'utenza, puoi anche gestire i tag utilizzando la console di Amazon Cognito.

Per suggerimenti sull'utilizzo dei tag, consulta il post AWSstrategie di assegnazione di tag nel blog AWS Risposte.

Nelle sezioni seguenti vengono fornite ulteriori informazioni sui tag per Amazon Cognito.

Risorse supportate in Amazon Cognito

Le seguenti risorse in Amazon Cognito supportano l'assegnazione di tag:

- Bacini d'utenza
- Pool di identità

Limitazioni applicate ai tag

Le seguenti restrizioni si applicano ai tag sulle risorse di Amazon Cognito:

- Numero massimo di tag che è possibile assegnare a una risorsa: 50
- Lunghezza massima della chiave: 128 caratteri Unicode
- Lunghezza massima del valore: 256 caratteri Unicode
- Caratteri validi per chiavi e valori: a-z, A-Z, 0-9, spazi e i seguenti caratteri: _ . : / = + - e @
- Per chiavi e valori viene fatta distinzione tra maiuscole e minuscole
- Non utilizzare `aws:` come prefisso per le chiavi; l'utilizzo di questo prefisso è esclusivo di AWS

Gestione dei tag tramite la console di Amazon Cognito

Puoi utilizzare la console Amazon Cognito per gestire i tag assegnati ai tuoi bacini d'utenza.

Per aggiungere tag a un bacino d'utenza

1. Passa alla console di Amazon Cognito. Se richiesto, inserisci le tue credenziali AWS.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo.
4. Scegli la scheda User pool properties (Proprietà del bacino d'utenza) e individua Tag.
5. Scegli Add tags (Aggiungi tag) per aggiungere il primo tag. Se hai precedentemente assegnato tag a questo bacino d'utenza, in Gestisci tag seleziona Aggiungi un altro.
6. Specifica i valori per Tag Key (Chiave tag) e Tag Value (Valore tag).
7. Per ogni ulteriore tag che desideri aggiungere, seleziona Add another (Aggiungi un altro).
8. Una volta completata l'aggiunta di tag, scegli Save changes (Salva modifiche).

Nella pagina Gestisci tag, si possono anche modificare le chiavi e i valori di ogni tag esistente. Per rimuovere un tag, scegli Remove (Rimuovi).

Esempi di AWS CLI

La AWS CLI offre dei comandi che si possono utilizzare per gestire i tag assegnati ai bacini d'utenza e ai pool di identità di Amazon Cognito.

Assegnazione di tag

Utilizza i seguenti comandi per assegnare tag ai bacini d'utenza e di identità esistenti.

Example Comando **`tag-resource`** per i bacini d'utenza

Assegna tag a un bacino d'utenza utilizzando il comando `tag-resource` all'interno del set di comandi `cognito-idp`.

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test
```

Questo comando include i seguenti parametri:

- `resource-arn`: l'Amazon Resource Name (ARN) del bacino d'utenza a cui stai applicando i tag. Per trovare l'ARN, seleziona il bacino d'utenza nella console Amazon Cognito e visualizza il valore ARN del pool nella scheda Impostazioni generali.
- `tags`: le coppie chiave-valore dei tag, nel formato *`key`*=*`value`*.

Per assegnare più tag in una sola volta, specificali in un elenco separato da virgole:

```
$ aws cognito-idp tag-resource \
> --resource-arn user-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Example Comando **`tag-resource`** per i pool di identità

Assegna tag a un pool di identità utilizzando il comando `tag-resource` all'interno del set di comandi `cognito-identity`.

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test
```

Questo comando include i seguenti parametri:

- `resource-arn`: l'Amazon Resource Name (ARN) del pool di identità a cui stai applicando i tag. Per cercare l'ARN, seleziona il pool di identità nella console Amazon Cognito e scegli Modifica pool di identità. Quindi, in Identity pool ID (ID pool di identità), seleziona Show ARN (Mostra ARN).

- `tags`: le coppie chiave-valore dei tag, nel formato *`key`*=*`value`*.

Per assegnare più tag in una sola volta, specificali in un elenco separato da virgole:

```
$ aws cognito-identity tag-resource \
> --resource-arn identity-pool-arn \
> --tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Visualizzazione dei tag

Utilizza i seguenti comandi per visualizzare i tag che hai assegnato ai bacini d'utenza e di identità.

Example Comando **`list-tags-for-resource`** per i bacini d'utenza

Visualizza i tag assegnati a un bacino d'utenza utilizzando il comando `list-tags-for-resource` all'interno del set di comandi `cognito-idp`:

```
$ aws cognito-idp list-tags-for-resource --resource-arn user-pool-arn
```

Example Comando **`list-tags-for-resource`** per i pool di identità

Visualizza i tag assegnati a un pool di identità utilizzando il comando `list-tags-for-resource` all'interno del set di comandi `cognito-identity`.

```
$ aws cognito-identity list-tags-for-resource --resource-arn identity-pool-arn
```

Rimozione dei tag

Utilizza i seguenti comandi per rimuovere tag dai bacini d'utenza e di identità.

Example Comando **`untag-resource`** per i bacini d'utenza

Rimuovi tag da un bacino d'utenza utilizzando il comando `untag-resource` all'interno del set di comandi `cognito-idp`:

```
$ aws cognito-idp untag-resource \
> --resource-arn user-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Per il parametro `--tag-keys`, specifica una o più chiavi di tag. Non includere i valori di tag. Separa le chiavi con spazi.

Example Comando **`untag-resource`** per i pool di identità

Rimuovi tag da un pool di identità utilizzando il comando `untag-resource` all'interno del set di comandi `cognito-identity`:

```
$ aws cognito-identity untag-resource \
> --resource-arn identity-pool-arn \
> --tag-keys Stage CostCenter Owner
```

Per il parametro `--tag-keys`, specifica una o più chiavi di tag. Non includere i valori di tag.

> **Important**
>
> Dopo aver eliminato un utente o un pool di identità, i tag correlati al pool eliminato possono ancora apparire nella console o nelle chiamate API fino a 30 giorni dopo l'eliminazione.

Applicazione di tag durante la creazione delle risorse

Utilizza i seguenti comandi per assegnare tag al momento della creazione di un bacino d'utenza o un pool di identità.

Example Comando **`create-user-pool`** con tag

Quando crei un bacino d'utenza utilizzando il comando `create-user-pool`, puoi specificare tag con il parametro `--user-pool-tags`:

```
$ aws cognito-idp create-user-pool \
> --pool-name user-pool-name \
> --user-pool-tags Stage=Test,CostCenter=80432,Owner=SysEng
```

Le coppie chiave-valore per i tag devono essere nel formato *`key`*=*`value`*. Se aggiungi più tag, specificali in un elenco separato da virgole.

Example Comando **`create-identity-pool`** con tag

Quando crei un pool di identità utilizzando il comando `create-identity-pool`, puoi specificare tag con il parametro `--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
```

Le coppie chiave-valore per i tag devono essere nel formato *key*=*value*. Se aggiungi più tag, specificali in un elenco separato da virgole.

Gestione dei tag tramite l'API di Amazon Cognito

Puoi utilizzare le seguenti operazioni nell'API di Amazon Cognito per gestire i tag dei bacini d'utenza e dei pool di identità.

Operazioni API per i tag del bacino d'utenza

Utilizza le seguenti operazioni API per assegnare, visualizzare e rimuovere tag per i bacini d'utenza.

- TagResource
- ListTagsForResource
- UntagResource
- CreateUserPool

Operazioni API per i tag del pool di identità

Utilizza le seguenti operazioni API per assegnare, visualizzare e rimuovere tag per i pool di identità.

- TagResource
- ListTagsForResource
- UntagResource
- CreateIdentityPool

Quote in Amazon Cognito

Amazon Cognito dispone di quote predefinite, precedentemente definite limiti, per il numero massimo di operazioni che possono essere eseguite nell'account. Amazon Cognito dispone anche di quote relative al numero massimo e alle dimensioni massime delle risorse di Amazon Cognito.

Ogni quota di Amazon Cognito rappresenta un volume massimo di richieste Regione AWS in una a una. Account AWS Ad esempio, le app possono effettuare richieste API fino alla frequenza di quota predefinita (RPS) per operazioni `UserAuthentication` rispetto a tutti i pool di utenti negli Stati Uniti orientali (Virginia settentrionale). Le tue app in Asia Pacifico (Tokyo) possono generare lo stesso volume di richieste per tutti i tuoi pool di utenti nella rispettiva regione. AWS può concedere una richiesta di aumento della quota solo in una regione alla volta. Un aumento della quota nella regione Stati Uniti orientali (Virginia settentrionale) non ha alcun effetto sulla frequenza massima di richiesta nella regione Asia Pacifico (Tokyo).

Argomenti

- Informazioni sulle quote di frequenza di richiesta API
- Gestione delle quote di frequenza di richiesta API
- Le operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste
- Frequenza delle richieste di operazione API dei pool di identità di Amazon Cognito (identità federate)
- Quote relative al numero e alla dimensione delle risorse

Informazioni sulle quote di frequenza di richiesta API

Categorizzazione delle quote

Amazon Cognito impone una frequenza massima di richieste per le operazioni API. Per ulteriori informazioni sulle operazioni API rese disponibili da Amazon Cognito, consulta Riferimenti endpoint e API di Amazon Cognito. Per i pool di utenti, queste operazioni sono raggruppate in categorie di casi d'uso comuni come `UserAuthentication` o `UserCreation`. Per un elenco delle operazioni API del pool di utenti per categoria, consulta. Le operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste

Nella console Service Quotas, puoi monitorare l'utilizzo delle quote per pool di utenti di categoria e pool di identità. Se il tasso di richieste dei tuoi pool di utenti Amazon Cognito supera o supera una

quota, puoi acquistare capacità aggiuntiva. Puoi tenere traccia dell'utilizzo delle quote del pool di utenti per categoria e degli aumenti delle quote di acquisto nella console Service Quotas.

Le quote di operazione sono definite come il numero massimo di richieste consentite al secondo (RPS) per tutte le operazioni all'interno di una categoria. Il servizio dei bacini d'utenza Amazon Cognito applica quote a tutte le operazioni in ciascuna categoria. Ad esempio, la categoria `UserCreation` include quattro operazioni: `SignUp`, `ConfirmSignUp`, `AdminCreateUser` e `AdminConfirmSignUp`. È assegnata con una quota combinata di 50 RPS. Se vengono eseguite più operazioni contemporaneamente, ogni operazione all'interno di questa categoria può richiamare fino a 50 RPS separate o combinate.

> **Note**
>
> Le quote di categoria si applicano solo a pool di utenti. Amazon Cognito applica ogni quota del pool di identità a una singola operazione. Per le quote di frequenza delle richieste, sia per categoria che per operazione, AWS misura il tasso aggregato di tutte le richieste provenienti da tutti i pool di utenti o pool di identità presenti in un'unica regione. Account AWS

Operazioni API del bacino d'utenza di Amazon Cognito con una gestione speciale della frequenza delle richieste

Le quote di operazione sono misurate e applicate per le richieste totali combinate a livello di categoria, ad eccezione delle operazioni `AdminRespondToAuthChallenge` e `RespondToAuthChallenge`, in cui vengono applicate regole di gestione speciali.

La `UserAuthentication` categoria include quattro operazioni nell'API dei pool di utenti di Amazon Cognito:`AdminInitiateAuth`, `InitiateAuthAdminRespondToAuthChallenge`, e. `RespondToAuthChallenge` Inoltre, l'autenticazione degli utenti nell'interfaccia utente ospitata contribuisce a questa quota. Le operazioni `InitiateAuth` e `AdminInitiateAuth` sono misurate e applicate per quota di categoria. Le operazioni di corrispondenza `RespondToAuthChallenge` e `AdminRespondToAuthChallenge` sono soggette a una quota separata che è tre volte superiore al limite di categoria `UserAuthentication`. Questa quota elevata soddisfa le molteplici sfide di autenticazione impostate nelle tue app. La quota è sufficiente a coprire la grande maggioranza dei casi d'uso. Dopo che l'app ha fornito fino a tre risposte ai problemi di autenticazione, le richieste aggiuntive vengono conteggiate ai fini della quota di `UserAuthentication` categoria. L'autenticazione a più fattori (MFA), l'autenticazione dei dispositivi e l'autenticazione personalizzata sono tutti esempi di richieste di sfida che potresti inserire nel tuo pool di utenti.

Ad esempio, se la tua quota per la `UserAuthentication` categoria è di 80 RPS, puoi chiamare `RespondToAuthChallenge` o `AdminRespondToAuthChallenge` a una velocità fino a 240 RPS (3* 80 RPS). Se il tuo pool di utenti richiede quattro round di sfida per autenticazione e 70 utenti accedono al secondo, il totale `RespondToAuthChallenge` è di 280 RPS (70 x 4), ovvero 40 RPS in più rispetto alla quota. Il valore di 40 RPS extra viene aggiunto a 70 chiamate `InitiateAuth`, rendendo l'utilizzo totale della categoria `UserAuthentication` 110 RPS (40 + 70). Poiché questo valore supera la quota di categoria fissata a 80 RPS per 30 RPS, Amazon Cognito limita le richieste provenienti dalla tua app.

Monthly active users (Utenti attivi mensili)

Quando Amazon Cognito calcola la fatturazione del pool di utenti, ti addebita una tariffa per ogni utente attivo mensile (MAU). Considera il numero di MAU attuale e previsto nella pianificazione delle richieste di aumento delle quote. Un utente viene conteggiato come MAU se, entro un mese di calendario, viene eseguita un'operazione di identità correlata a tale utente. Le attività che rendono attivo un utente includono le seguenti.

- Registrazione o creazione amministrativa di un utente
- Accesso
- Disconnettersi
- Conferma dell'account utente o verifica degli attributi
- Reimpostazione della password
- Modifica degli attributi utente, della sottoscrizione di ruppo o delle preferenze MFA
- Esecuzione di query degli attributi dettagliati di un utente
- Attivazione, disattivazione o eliminazione dell'utente

> **Note**
>
> La categoria Interroga gli attributi dettagliati di un utente include il funzionamento dell'API AdminGetUser, ma non ListUsers. Un' user-by-user interrogazione dettagliata in un ampio pool di utenti può avere un impatto significativo sulla AWS fattura. Per evitare addebiti eccessivi, raccogli i dati degli utenti `ListUsers` o archivia le informazioni sugli utenti in un database esterno.

Gestione delle quote di frequenza di richiesta API

Identificazione dei requisiti delle quote

> **Important**
>
> Se aumenti le quote di Amazon Cognito per categorie come`UserAuthentication`, oppure `UserCreationAccountRecovery`, potresti dover aumentare le quote per altre. Servizi AWS Ad esempio, i messaggi inviati da Amazon Cognito con Amazon Simple Notification Service (Amazon SNS) e Amazon Simple Email Service (Amazon SES) possono non riuscire se le quote di richiesta sono insufficienti in tali servizi.

Per calcolare i requisiti delle quota, determina quanti utenti attivi interagiranno con l'applicazione in un determinato periodo di tempo. Ad esempio, se prevedi l'accesso medio di un milione di utenti attivi in un periodo di otto ore, sarà necessario essere in grado di autenticare una media di 35 utenti al secondo.

Inoltre, se si suppone che la sessione utente media sia di due ore e che i token siano configurati per scadere dopo un'ora, durante questa sessione di due ore ogni utente dovrà aggiornare una volta i propri token. La quota media richiesta per la categoria `UserAuthentication` per supportare questo carico è 70 RPS.

Se si presuppone un peak-to-average rapporto di 3:1 tenendo conto della variazione della frequenza di accesso degli utenti durante il periodo di otto ore, è necessaria la quota desiderata di 200 RPS. `UserAuthentication`

> **Note**
>
> Se si chiamano più operazioni per ogni azione utente, dovrai sommare i tassi di chiamata delle singole operazioni a livello di categoria.

Ottimizza le frequenze di richiesta per i limiti di quota

Poiché l'aumento dei limiti tariffari delle API comporta costi aggiuntivi AWS , valuta la possibilità di apportare modifiche al modello di utilizzo prima di richiedere un aumento della quota. Di seguito sono riportati alcuni esempi di architettura di app che ottimizza i tassi di richiesta.

Nuovo tentativo dopo un periodo di attesa di back-off

È possibile rilevare gli errori con ogni chiamata API e quindi riprovare dopo un periodo di back-off. È possibile regolare l'algoritmo di back-off in base alle esigenze aziendali e al carico. Gli SDK di Amazon dispongono di una logica di tentativi incorporata. Per ulteriori informazioni, consulta Strumenti su AWS cui basarsi.

Utilizzo di un database esterno per gli attributi aggiornati di frequente

Se l'applicazione richiede diverse chiamate a un bacino d'utenza per leggere o scrivere attributi personalizzati, utilizza l'archiviazione esterna. Puoi utilizzare il tuo database preferito per archiviare attributi personalizzati oppure utilizzare un livello di cache per caricare un profilo utente durante l'accesso. Puoi fare riferimento a questo profilo dalla cache quando necessario invece di ricaricare il profilo utente da un bacino d'utenza.

Convalida i token web JSON (JWT) sul lato client

Le applicazioni devono convalidare i token JWT prima di potersi fidare di loro. Puoi verificare la firma e la validità dei token sul lato client senza inviare richieste API a un pool di utenti. Una volta convalidato il token, sarà possibile considerare attendibili le attestazioni nel token e utilizzare le attestazioni invece di creare più chiamate API `getUser`. Per ulteriori informazioni, consulta Verifica di un token Web JSON.

Limitare il traffico verso la tua applicazione Web con una sala d'attesa

Se prevedi traffico da un numero elevato di utenti che effettuano l'accesso durante un evento limitato nel tempo, come un esame o un evento live, puoi ottimizzare il traffico delle richieste con meccanismi di limitazione automatica. Ad esempio, puoi impostare una sala d'attesa in cui gli utenti possono attendere fino a quando non è disponibile una sessione, consentendoti di elaborare le richieste quando la capacità è disponibile. Consulta Soluzione Virtual Waiting Room AWS per un'architettura di riferimento di una sala d'attesa.

JWT nella cache

Riutilizza i token di accesso fino alla loro scadenza. Per un esempio di framework con memorizzazione nella cache dei token in un API Gateway, vedereCaching dei token. Invece di generare richieste API per interrogare le informazioni degli utenti, memorizzate nella cache i token ID fino alla loro scadenza e leggete gli attributi utente dalla cache.

Per ulteriori informazioni sull'utilizzo delle frequenze di richiesta API in AWS, consulta Gestione e monitoraggio della limitazione delle API nei carichi di lavoro. Per informazioni sull'ottimizzazione delle operazioni di Amazon Cognito che aggiungono costi alla bolletta, AWS consulta. Gestione dei costi

Monitoraggio dell'uso delle quote

Amazon Cognito genera `CallCount` e `ThrottleCount` misura in Amazon CloudWatch per ogni categoria di operazioni API a livello di account. Puoi utilizzare `CallCount` per tenere traccia del numero totale di chiamate effettuate dai clienti in relazione a una categoria. Puoi utilizzare `ThrottleCount` per tenere traccia del numero totale di chiamate limitate relative a una categoria. Puoi utilizzare i parametri `CallCount` e `ThrottleCount` con la statistica `Sum` per contare il numero totale di chiamate in una categoria. Per ulteriori informazioni, consulta le metriche di CloudWatch utilizzo.

Quando si monitorano le quote di servizio, l'utilizzo è la percentuale di una quota di servizio in uso. Ad esempio, se il valore della quota è 200 risorse e 150 risorse sono in uso, l'utilizzo è 75%. L'uso è invece il numero di risorse o operazioni in uso per una quota di servizio.

Monitoraggio dell'utilizzo tramite metriche CloudWatch

Puoi tracciare e raccogliere le metriche di utilizzo dei pool di utenti di Amazon Cognito con. CloudWatch La CloudWatch dashboard mostra le metriche relative a tutti Servizio AWS gli elementi utilizzati. Con CloudWatch, puoi creare allarmi metrici per avvisarti o modificare una risorsa specifica che stai monitorando. Per ulteriori informazioni sulle CloudWatch metriche, consulta Tieni traccia delle metriche di utilizzo CloudWatch .

Monitoraggio dell'utilizzo tramite i parametri Service Quotas

I pool di utenti di Amazon Cognito sono integrati con Service Quotas, un'interfaccia console per visualizzare e gestire l'utilizzo delle quote di servizio. Nella console Service Quotas, puoi cercare il valore di una quota specifica, visualizzare le informazioni di monitoraggio, richiedere un aumento della quota o impostare CloudWatch allarmi. Dopo un certo periodo di attività dell'account, puoi visualizzare un grafico dell'utilizzo delle risorse.

La colonna Valore di quota a livello di account applicato nella console Service Quotas per i pool di utenti di Amazon Cognito e i pool di identità di Amazon Cognito mostra la tua quota attuale. La colonna Utilizzo mostra il tasso attuale di utilizzo delle quote. Le quote regolabili dei pool di utenti di Amazon Cognito requests-per-second (RPS) mostrano il loro utilizzo corrente. La console Service Quotas può anche accedere alle CloudWatch metriche per esaminare più da vicino una metrica di quota selezionata. Per ulteriori informazioni sulla visualizzazione delle quote nella console Service Quotas, consulta Visualizzazione di Service Quotas.

Tieni traccia degli utenti attivi mensili (MAU)

Il numero di utenti attivi mensili (MAU) nel tuo pool di utenti fornisce dati importanti per pianificare l'aumento delle quote relative alle richieste. Puoi confrontare i tassi di richiesta API con il numero di utenti attivi in un determinato periodo di tempo. Con queste conoscenze, potete calcolare in che modo un aumento degli utenti attivi delle vostre applicazioni influirà sulle quote del vostro modello di utilizzo. Ad esempio, immaginate che le vostre applicazioni combinate negli Stati Uniti occidentali (Oregon) generino 2 milioni di utenti attivi in un mese e che la vostra `UserAuthentication` categoria riceva occasionalmente errori di limitazione rispetto alla quota predefinita di 120 richieste al secondo (RPS). Nel mese precedente, prima del successo della vostra campagna pubblicitaria, avevate 1 milione di MAU e le vostre applicazioni non superavano mai gli 80 RPS. Se prevedi un picco simile a seguito di un nuovo spot televisivo, potresti acquistare altri 40 RPS per soddisfare il prossimo milione di utenti con una quota adeguata di 160 RPS.

Per recensire i tuoi MAU

Accedi alla AWS Billing console e rivedi una fattura recente. Nella sezione Addebiti per servizio, puoi filtrare su Cognito per visualizzare un'analisi dettagliata delle tue MAU per quel periodo di fatturazione.

Richiesta di aumento delle quote

Amazon Cognito prevede una quota per il numero massimo di operazioni al secondo che puoi eseguire nei pool di utenti e nei pool di identità di ciascuno. Regione AWS Puoi acquistare un aumento delle quote di richiesta API dei pool di utenti di Amazon Cognito regolabili. Controlla la tua quota attuale e acquista un aumento dalla console Service Quotas o con le operazioni dell'API Service Quotas e. `ListAWSDefaultServiceQuotas` `RequestServiceQuotaIncrease`

- Per acquistare un aumento della quota utilizzando la console Service Quotas, consulta Richiesta di un aumento della quota API nella Guida per l'utente di Service Quotas.
- AWS prevede il completamento delle richieste di aumento delle quote entro 10 giorni. Tuttavia, diverse considerazioni potrebbero far sì che il tempo di elaborazione della richiesta superi i 10 giorni. Alcune richieste, ad esempio, potrebbero richiedere che Amazon Cognito fornisca capacità hardware aggiuntiva e gli aumenti stagionali dei volumi di richieste potrebbero causare ritardi.
- Se la quota non è disponibile in Service Quotas, utilizza il modulo di incremento dei limiti di servizio.

> **Important**
>
> Solo le quote modificabili possono essere aumentate. È necessario acquistare una maggiore capacità di quota. Per l'aumento dei prezzi delle quote, consulta i prezzi di Amazon Cognito.

Le operazioni API del bacino d'utenza di Amazon Cognito e le quote di frequenza delle richieste

Poiché Amazon Cognito presenta classi di operazioni API sovrapposte con modelli di autorizzazione diversi, ogni operazione appartiene a una categoria. Ogni categoria ha la propria quota in pool per tutte le operazioni API del membro, in tutti i pool di utenti in una Regione AWS nell'account. È possibile richiedere solo l'incremento a quote di categoria regolabili. Per ulteriori informazioni, consulta Richiesta di aumento delle quote. Gli adeguamenti della quota si applicano ai pool di utenti nell'account di una singola regione. Amazon Cognito limita le operazioni in alcune categorie[3] a 5 richieste al secondo (RPS), per pool di utenti. La quota predefinita (RPS) si applica inoltre a tutti i pool di utenti in un. Account AWS

> **Note**
>
> La quota per ogni categoria viene misurata in Utenti attivi mensili (MAU). Account AWS con meno di due milioni di MAU possono funzionare entro la quota predefinita. Se disponi di meno di un milione di MAU e Amazon Cognito limita le richieste, valuta la possibilità di ottimizzare la tua app. Per ulteriori informazioni, consulta Ottimizza le frequenze di richiesta per i limiti di quota.

Le quote delle operazioni di categoria vengono applicate a tutti gli utenti in tutti i pool di utenti all'interno di una Regione AWS. Amazon Cognito mantiene anche una quota per il numero di richieste che l'app può generare rispetto a un singolo utente. È necessario limitare le richieste API per utente come mostrato nella tabella seguente.

Quote frequenza di richiesta per utente di pool di utenti di Amazon Cognito

Operazione	Operazioni per utente al secondo
Lettura profilo utente	10

Operazione	Operazioni per utente al secondo
Esempi: `GetUser`, `GetDevice`	
Scrittura profilo utente Esempi: `UpdateUserAttributes` , `SetUserSettings`	10

È necessario limitare le richieste API per categoria come mostrato nella tabella seguente.

Quote frequenza di richiesta per categoria di pool di utenti di Amazon Cognito

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserAuthentication` • InitiateAuth • Aggiornamento del token con `InitiateAuth` o Endpoint Token • RespondToAuthChallenge[1] • AdminInitiateAuth • AdminRespondToAuthChallenge[1] • Accesso all'interfaccia utente ospitata e MFA in codice di autorizzazione o concessioni implicite[2]	Operazioni che autenticano (accesso) un utente. Queste operazioni sono soggette a Operazioni API del bacino d'utenza di Amazon Cognito con una gestione speciale della frequenza delle richieste .	120	Sì

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserCreation` • SignUp • ConfirmSignUp • AdminCreateUser • AdminConfirmSignUp	Operazioni che creano o confermano un utente locale Amazon Cognito. Si tratta di un utente creato e verificato direttamente dai bacini d'utenza di Amazon Cognito.	50	Sì
`UserFederation` Operazioni che federano (autenticano) gli utenti con un provider di identità di terze parti nei bacini d'utenza Amazon Cognito.	Operazioni che inviano una risposta IdP a un endpoint di federazione di pool di utenti. Le operazioni OIDC o del provider social che generano un token IdP e tutte le richieste SAML contribuiscono a questa quota.	25	Sì

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserAccountRecovery` • ChangePassword • ConfirmForgotPassword • ForgotPassword • AdminResetUserPassword • AdminSetUserPassword • RespondToAuthChallenge[1] • AdminRespondToAuthChallenge[1] • Reimpostazione della password dell'interfaccia utente ospitata	Operazioni che recuperano l'account di un utente o modificano o aggiornano la password di un utente.	30	No
`UserRead` • AdminGetUser • GetUser	Operazioni che recuperano un utente dai bacini d'utenza.	120	Sì

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserUpdate` • AdminAddUserToGroup • AdminDeleteUserAttributes • AdminUpdateUserAttributes • AdminDeleteUser • AdminDisableUser • AdminEnableUser • AdminLinkProviderForUser • AdminDisableProviderForUser • VerifyUserAttribute • DeleteUser • DeleteUserAttributes • UpdateUserAttributes • AdminUserGlobalSignOut • GlobalSignOut • AdminRemoveUserFromGroup	Operazioni utilizzi per gestire gli utenti e gli attributi utente.	25	No
`UserToken` • RevokeToken	Operazioni per la gestione dei token	120	Sì

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserResourceRead` • AdminGetDevice • AdminListGroupsForUser • AdminListDevices • GetDevice • ListDevices • GetUserAttributeVerificationCode • ResendConfirmationCode • AdminListUserAuthEvents	Operazioni che recuperano informazioni sulle risorse utente da Amazon Cognito, ad esempio un dispositivo memorizzato o l'appartenenza a un gruppo di utenti.	50	Sì

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserResourceUpdate` • AdminForgetDevice • AdminUpdateAuthEventFeedback • AdminSetUserRiferimento MFAP • AdminSetUserSettings • AdminUpdateDeviceStatus • UpdateDeviceStatus • UpdateAuthEventFeedback • ConfirmDevice • SetUserRiferimentoMFAP • SetUserSettings • VerifySoftwareToken • AssociateSoftwareToken • ForgetDevice	Operazioni che aggiornano informazioni sulle risorse per un utente, ad esempio un dispositivo memorizzato o l'appartenenza a un gruppo di utenti.	25	No
`UserList` • ListUsers • ListUsersInGroup	Operazioni che restituiscono un elenco di utenti.	30	No

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserPoolRead` • DescribeUserPool • ListUserPools	Operazioni che leggono i bacini d'utenza.	15	No
`UserPoolUpdate` • CreateUserPool • UpdateUserPool • DeleteUserPool	Operazioni che creano, aggiornano o eliminano i bacini d'utenza.	15	No

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserPoolResourceRead` • DescribeIdentityProvider • DescribeResourceServer • DescribeUserImportJob • DescribeUserPoolDomain • GetCSVHeader • GetGroup • GetSigningCertificate • GetIdentityProviderByIdentifier • GetUserPoolMfaConfig • ListGroups • ListIdentityProviders • ListResourceServers • ListTagsForResource • ListUserImportJobs • DescribeRiskConfiguration • GetUICustomization	Operazioni che recuperano informazioni sulle risorse, come gruppi o server di risorse, da un pool di utenti.[3]	20	No

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
`UserPoolResourceUpdate` • AddCustomAttributes • CreateGroup • CreateIdentityProvider • CreateResourceServer • CreateUserImportJob • CreateUserPoolDomain • DeleteGroup • DeleteIdentityProvider • DeleteResourceServer • DeleteUserPoolDomain • SetUserPoolMfaConfig • StartUserImportJob • StopUserImportJob • UpdateGroup • UpdateIdentityProvider • UpdateResourceServer	Operazioni che modificano le risorse, come gruppi o server di risorse, in un pool di utenti.[3]	15	No

Categoria	Descrizione	Quota predefinita (RPS)	Regolabile
• UpdateUserPoolDomain • SetRiskConfiguration • SetUICustomization • TagResource • UntagResource			
`UserPoolClientRead` • DescribeUserPoolClient • ListUserPoolClients	Operazioni che recuperano informazioni sui client del pool di utenti.[3]	15	No
`UserPoolClientUpdate` • CreateUserPoolClient • DeleteUserPoolClient • UpdateUserPoolClient	Operazioni che creano, aggiornano ed eliminano i client del pool di utenti.[3]	15	No
`ClientAuthentication` Richieste di tipo di concessione `client_credentials` all'endpoint token.	Operazioni che generano credenziali da utilizzare nelle richieste di autorizzazione machine-to-machine	150	No

[1] A `RespondToAuthChallenge` o una `AdminRespondToAuthChallenge` risposta con un `ChallengeName` di `NEW_PASSWORD_REQUIRED` conta ai fini della `UserAccountRecovery` categoria. Tutte le altre risposte alla sfida vengono conteggiate ai fini della `UserAuthentication` categoria.

[2] Ogni operazione dell'interfaccia utente ospitata durante l'accesso contribuisce con una richiesta alla quota. Ad esempio, un utente che accede e fornisce un codice MFA contribuisce con 2 richieste. Il riscatto dei token nell'ambito della concessione di codici di autorizzazione è soggetto all'assegnazione di quote aggiuntive alla stessa aliquota prevista per la categoria. `UserAuthentication`

[3] Ogni singola operazione in questa categoria presenta un vincolo che impedisce che l'operazione venga richiamata a una velocità superiore a 5 RPS per un singolo pool di utenti.

Frequenza delle richieste di operazione API dei pool di identità di Amazon Cognito (identità federate)

Operazione	Descrizione	Quota predefinita (RPS)[1]	Regolabile	Idoneità all'incremento delle quote
`GetId`	Recupera un ID di identità da un pool di identità.	25	Sì	Contatta il team del tuo account.
`GetOpenIdToken`	Recupera un token OpenID da un pool di identità nel flusso di lavoro classico.	200	Sì	Contatta il team del tuo account.
`GetCredentialsForIdentity`	Recupera AWS le credenziali da un pool di identità nel	200	Sì	Contatta il team del tuo account.

Operazione	Descrizione	Quota predefinita (RPS)[1]	Regolabile	Idoneità all'incremento delle quote
	flusso di lavoro avanzato.			
`GetOpenIdTokenForDeveloperIdentity`	Recupera un token OpenID da un pool di identità nel flusso di lavoro per gli sviluppatori.	50	Sì	Contatta il team del tuo account.
`ListIdentities`	Recupera un elenco di ID identità in un pool di identità.	5	Sì	Contatta il team del tuo account.
`DeleteIdentities`	Elimina una o più identità registrate da un pool di identità.	10	Sì	Contatta il team del tuo account.
`TagResource`	Applica un tag a un pool di identità.	5	Sì	Contatta il team del tuo account.
`UntagResource`	Rimuovi un tag da un pool di identità.	5	Sì	Contatta il team del tuo account.
`ListTagsForResource`	Visualizza un elenco dei tag applicati a un pool di identità.	10	Sì	Contatta il team del tuo account.

[1] La quota predefinita è la quota minima del tasso di richieste per i pool di identità di qualsiasi Regione AWS paese dell'Unione. Account AWS La quota RPS potrebbe essere più alta in alcune regioni.

Quote relative al numero e alla dimensione delle risorse

Le quote delle risorse sono il numero o la dimensione massima di risorse, campi di input, durata temporale e altre funzionalità varie in Amazon Cognito.

È possibile richiedere un adeguamento di alcune quote delle risorse nella console Service Quotas o da un modulo di incremento dei limiti di servizio. Per richiedere un aumento delle quote utilizzando la console Service Quotas, consulta Richiesta di aumento di una quot nella Guida per l'utente di Service Quotas. Se la quota non è disponibile in Service Quotas, utilizza il modulo di incremento dei limiti di servizio.

> **Note**
>
> Le quote di risorse a Account AWS livello, ad esempio i pool di utenti per regione, si applicano alle risorse di Amazon Cognito in ciascuna regione. Regione AWS Ad esempio, si potrebbe disporre di 1.000 pool di utenti negli Stati Uniti orientali (Virginia settentrionale) e altri 1.000 in Europa (Stoccolma).

Le tabelle seguenti indicano le quote delle risorse predefinite e se sono regolabili.

Quote di risorse del bacino d'utenza di Amazon Cognito

Risorsa	Quota	Regolabile	Quota massima
Client app per bacino d'utenza	1.000	Sì	10.000
Pool di utenti per regione	1.000	Sì	10.000
Provider di identità per pool di utenti	300	Sì	1.000

Risorsa	Quota	Regolabile	Quota massima
Server delle risorse per pool di utenti	25	Sì	300
Utenti per bacino d'utenza	40.000.000	Sì	Contatta il team del tuo account.
Modifiche combinate totali nel trigger Lambda di pre-generazione di token [1]	5.000	Sì	Contatta il team del tuo account.
Attributi personalizzati per bacino d'utenza	50	No	N/D
Caratteri per attributo	2.048 byte	No	N/D
Caratteri per il nome di un attributo personalizzato	20	No	N/D
Caratteri minimi richiesti per la password nella policy per la password	6–99	No	N/D
Messaggi di posta elettronica inviati giornalmente per Account AWS [2]	50	No	N/D
Caratteri nell'oggetto dell'e-mail	140	No	N/D
Caratteri nel messaggio e-mail	20.000	No	N/D

Risorsa	Quota	Regolabile	Quota massima
Caratteri nel messaggio di verifica SMS	140	No	N/D
Caratteri nella password	256	No	N/D
Caratteri per il nome del provider di identità	32	No	N/D
Identificatori per provider di identità	50	No	N/D
Identità collegate a un utente	5	No	N/D
URL di callback per client di app	100	No	N/D
URL di disconnessione per client di app	100	No	N/D
Ambiti per server di risorse	100	No	N/D
Ambiti per client di app	50	No	N/D
Domini personalizzati per account	4	No	N/D
Gruppi a cui ogni utente può appartenere	100	No	N/D
Gruppi per pool di utenti	10.000	No	N/D

[1] Questa quota potrebbe essere riscontrata nei token di un Trigger Lambda di pre-generazione del token. Il numero di attestazioni esistenti e aggiunte, più gli ambiti nei token di accesso e di identità, deve sommarsi a un numero inferiore o uguale a tale quota. Le attestazioni e gli ambiti soppressi non contribuiscono a questa quota.

[2] Questa quota vale solo se utilizzi la funzionalità e-mail predefinita per un pool di utenti di Amazon Cognito. Per un volume di consegna e-mail più elevato, configura il pool di utenti in modo da utilizzare la configurazione e-mail di Amazon SES. Per ulteriori informazioni, consulta Impostazioni e-mail per i bacini d'utenza di Amazon Cognito.

Parametri di validità della sessione dei bacini d'utenza di Amazon Cognito

Token	Quota
Token ID	5 minuti - 1 giorno
Token di aggiornamento	1 ora - 3.650 giorni
Token di accesso	5 minuti - 1 giorno
Cookie di sessione per l'interfaccia utente ospitata	1 ora
Token di sessione di autenticazione	3 minuti - 15 minuti

Quote di risorse di sicurezza del codice dei pool di utenti di Amazon Cognito (non modificabili)

Risorsa	Quota
Periodo di validità del codice di conferma della registrazione	24 ore
Periodo di validità del codice di verifica dell'attr ibuto utente	24 ore
Periodo di validità del codice dell'autenticazione a più fattori (MFA)	3–15 minuti

Risorsa	Quota
Periodo di validità del codice della password dimenticata	1 ora
Numero massimo di richieste `ConfirmForgotPassword` e `ForgotPassword` per utente all'ora [1]	5-20
Numero massimo di richieste `ResendConfirmationCode` per utente all'ora	5
Numero massimo di richieste `ConfirmSignUp` per utente all'ora	15
Numero massimo di richieste `ChangePassword` per utente all'ora	5
Numero massimo di richieste `GetUserAttributeVerificationCode` per utente all'ora	5
Numero massimo di richieste `VerifyUserAttribute` per utente all'ora	15

[1] Amazon Cognito valuta i fattori di rischio nella richiesta di aggiornamento delle password e assegna una quota collegata al livello di rischio valutato. Per ulteriori informazioni, consulta Come comportarsi in caso di password dimenticata.

Quote delle risorse dei processi di importazione utenti del pool di utenti di Amazon Cognito

Risorsa	Quota	Regolabile	Quota massima
Processi di importazione utente per pool di utenti	1.000	Sì	Contatta il team del tuo account.

Risorsa	Quota	Regolabile	Quota massima
Numero massimo di caratteri per riga CSV di importazione utente	16,000	No	N/D
Dimensione massima dei file CSV	100 MB	No	N/D
Numero massimo di utenti per file CSV	500.000	No	N/D

Quote di risorsa per i pool di identità di Amazon Cognito (identità federate)

Risorsa	Quota	Regolabile	Quota massima
Pool di identità per account	1.000	Sì	N/D
Provider del bacino d'utenza di Amazon Cognito per pool di identità	50	Sì	1000
Lunghezza dei caratteri per il nome del pool di identità	128 byte	No	N/D
Lunghezza dei caratteri per il nome del provider di accesso	2.048 byte	No	N/D
Identità per pool di identità	Illimitato	No	N/D
Provider di identità per i quali è possibile	10	No	N/D

Risorsa	Quota	Regolabile	Quota massima
specificare mappature di ruoli			
Risultati da una singola chiamata di ricerca o elenco	60	No	N/D
Regole di controllo accessi basato sui ruoli (RBAC)	25	No	N/D

Quote di risorse Amazon Cognito Sync

Risorsa	Quota	Regolabile	Quota massima
Set di dati per identità	20	Sì	Contatta il team del tuo account.
Record per set di dati	1,024	Sì	Contatta il team del tuo account.
Dimensione di un singolo set di dati	1 MB	Sì	Contatta il team del tuo account.
Caratteri nel nome del set di dati	128 byte	No	N/D
Attesa per una pubblicazione in blocco, dopo una richiesta eseguita correttamente	24 ore	No	N/D

Riferimenti endpoint e API di Amazon Cognito

I seguenti riferimenti API descrivono gli endpoint del servizio per ogni funzionalità di Amazon Cognito. I pool di utenti Amazon Cognito dispongono delle seguenti opzioni: endpoint del pool di utenti con un dominio del pool di utenti e API dei pool di utenti. Per un'analisi dettagliata delle classi di operazioni API con l'API dei pool di utenti dei pool di utenti Amazon Cognito, consulta Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti.

Per un elenco degli endpoint del servizio per l'API dei pool di utenti per Regione AWS, consulta Endpoint del servizio nel Riferimento generale AWS.

Argomenti

- Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata
- Documentazione di riferimento API dei pool di utenti di Amazon Cognito
- Documentazione di riferimento dell'API dei pool di identità di Amazon Cognito (identità federate)
- Documentazione di riferimento dell'API di Amazon Cognito sync

Documentazione di riferimento degli endpoint di federazione del pool di utenti e dell'interfaccia utente ospitata

Amazon Cognito attiva le pagine Web pubbliche elencate qui quando si assegna un dominio al pool di utenti. Il dominio funge da punto di accesso centrale per tutti i client dell'app. Includono l'interfaccia utente ospitata, a cui gli utenti possono registrarsi e accedere (Endpoint Login) e da cui possono disconnettersi (Endpoint Logout). Per ulteriori informazioni su queste risorse, vedere Configurazione e utilizzo dell'interfaccia utente ospitata di Amazon Cognito e degli endpoint di federazione.

Queste pagine includono anche le risorse Web pubbliche che consentono al pool di utenti di comunicare con provider di identità SAML, OpenID Connect (OIDC) e OAuth 2.0 di terze parti (). IdPs Per eseguire l'accesso come un utente con un provider di identità federato, gli utenti devono avviare una richiesta all'interfaccia utente ospitata interattiva Endpoint Login o a Endpoint Authorize OIDC. L'endpoint Autorizza reindirizza gli utenti all'interfaccia utente ospitata o alla pagina di accesso IdP.

L'app può anche effettuare l'accesso come utenti locali con l'API dei pool di utenti Amazon Cognito. Un utente locale esiste esclusivamente nella directory del pool di utenti senza federazione tramite un IdP esterno.

Oltre all'interfaccia utente ospitata e agli endpoint federativi, Amazon Cognito si integra con gli SDK per Android, iOS JavaScript e altri. Gli SDK forniscono strumenti per eseguire operazioni API del pool di utenti con endpoint del servizio dell'API Amazon Cognito. Per ulteriori informazioni su questi endpoint del servizio, consulta Endpoint e quote di Amazon Cognito Identity.

> **Warning**
>
> Non aggiungere certificati Transport Layer Security (TLS) di entità finale o intermedi per i domini Amazon Cognito. AWS gestisce tutti i certificati per tutti gli endpoint e i domini con prefisso del pool di utenti. Le autorità di certificazione (CA) della catena di attendibilità che supporta i certificati Amazon Cognito si alternano e si rinnovano dinamicamente. Quando aggiungi la tua app a un certificato intermedio o principale, l'app può fallire senza preavviso durante la rotazione dei certificati. AWS
> Associa invece la tua applicazione a tutti i certificati root Amazon disponibili. Per ulteriori informazioni, consulta le best practice e i suggerimenti in Associazione dei certificati nella Guida per l'utente di AWS Certificate Manager .

Argomenti

- Documentazione di riferimento degli endpoint dell'interfaccia utente ospitata
- Documentazione di riferimento degli endpoint di federazione OAuth 2.0, OpenID Connect e SAML 2.0
- Concessioni OAuth 2.0
- Utilizzo di PKCE nelle concessioni di codici di autorizzazione con pool di utenti Amazon Cognito
- Risposte agli errori di federazione e dell'interfaccia utente ospitata

Documentazione di riferimento degli endpoint dell'interfaccia utente ospitata

Amazon Cognito attiva gli endpoint dell'interfaccia utente ospitata in questa sezione quando si aggiunge un dominio al pool di utenti. Sono pagine Web in cui gli utenti possono completare le operazioni di autenticazione di base di un pool di utenti. Includono pagine per la gestione delle password, l'autenticazione a più fattori (MFA) e la verifica degli attributi. Per ulteriori informazioni sull'esperienza utente nell'interfaccia utente ospitata, consultare Registrazione e accesso con l'interfaccia utente ospitata.

Le pagine Web che compongono l'interfaccia utente ospitata sono un'applicazione Web front-end per sessioni utente interattive con i tuoi clienti. L'app deve richiamare l'interfaccia utente ospitata nei browser degli utenti. Amazon Cognito non supporta l'accesso programmatico alle pagine Web di questo capitolo. Gli endpoint di federazione in Documentazione di riferimento degli endpoint di federazione OAuth 2.0, OpenID Connect e SAML 2.0 che restituiscono una risposta JSON possono essere interrogati direttamente nel codice dell'app. Endpoint Authorize viene reindirizzato all'interfaccia utente ospitata o a una pagina di accesso IdP e deve inoltre essere aperto nei browser degli utenti.

Gli argomenti di questa guida descrivono in dettaglio gli endpoint dell'interfaccia utente ospitata utilizzati di frequente. Amazon Cognito rende disponibili le pagine Web seguenti quando si assegna un dominio al pool di utenti.

Endpoint dell'interfaccia utente ospitata

URL dell'endpoint	Descrizione	Modalità di accesso
https://*dominio del pool di utenti*/login	Accede agli utenti locali e federati del pool di utenti.	Esegui il reindirizzamento da endpoint come Endpoint Authorize, `/logout` e `/confirmforgotPassword` . Per informazioni, consulta Endpoint Login.
https://*dominio del pool di utenti*/logout	Disconnette gli utenti del pool di utenti.	Collegamento diretto. Per informazioni, consulta Endpoint Logout.
https://*dominio pool di utenti*/confirmUser	Conferma gli utenti che hanno selezionato un collegamento e-mail per verificare il proprio account utente.	Collegamento selezionato dall'utente in un messaggio di posta elettronica.
https://*dominio del pool di utenti*/signup	Registra un nuovo utente. La pagina `/login` indirizza l'utente a `/signup` quando si seleziona Sign up (Registrati).	Collegamento diretto con gli stessi parametri di `/oauth2/authorize` .

URL dell'endpoint	Descrizione	Modalità di accesso
https://*dominio pool di utenti*/confirm	Dopo che il pool di utenti invia un codice di conferma a un utente che ha effettuato la registrazione, richiede il codice all'utente.	Solo reindirizzamento da `/signup`.
https://*dominio del pool di utenti*/forgotPassword	Richiede all'utente il nome utente e invia un codice di ripristino della password. La pagina `/login` indirizza l'utente a `/forgotPassword` quando si seleziona Forgot your password? (Hai dimenticato la password?).	1. Dal collegamento Password dimenticata a `/login`. 2. Collegamento diretto con gli stessi parametri di `/oauth2/authorize` .
https://*dominio pool di utenti*/confirmforgotPassword	Richiede all'utente il codice di ripristino della password e una nuova password. La pagina `/forgotPassword` indirizza l'utente a `/confirmforgotPassword` quando si seleziona Reset your password (Reimposta la password).	Solo reindirizzamento da `/forgotPassword` .
https://*dominio pool di utenti*/resendcode	Invia un nuovo codice di conferma a un utente che ha effettuato la registrazione nel pool di utenti.	Solo reindirizzamento dal collegamento Invia un nuovo codice a `/confirm`.

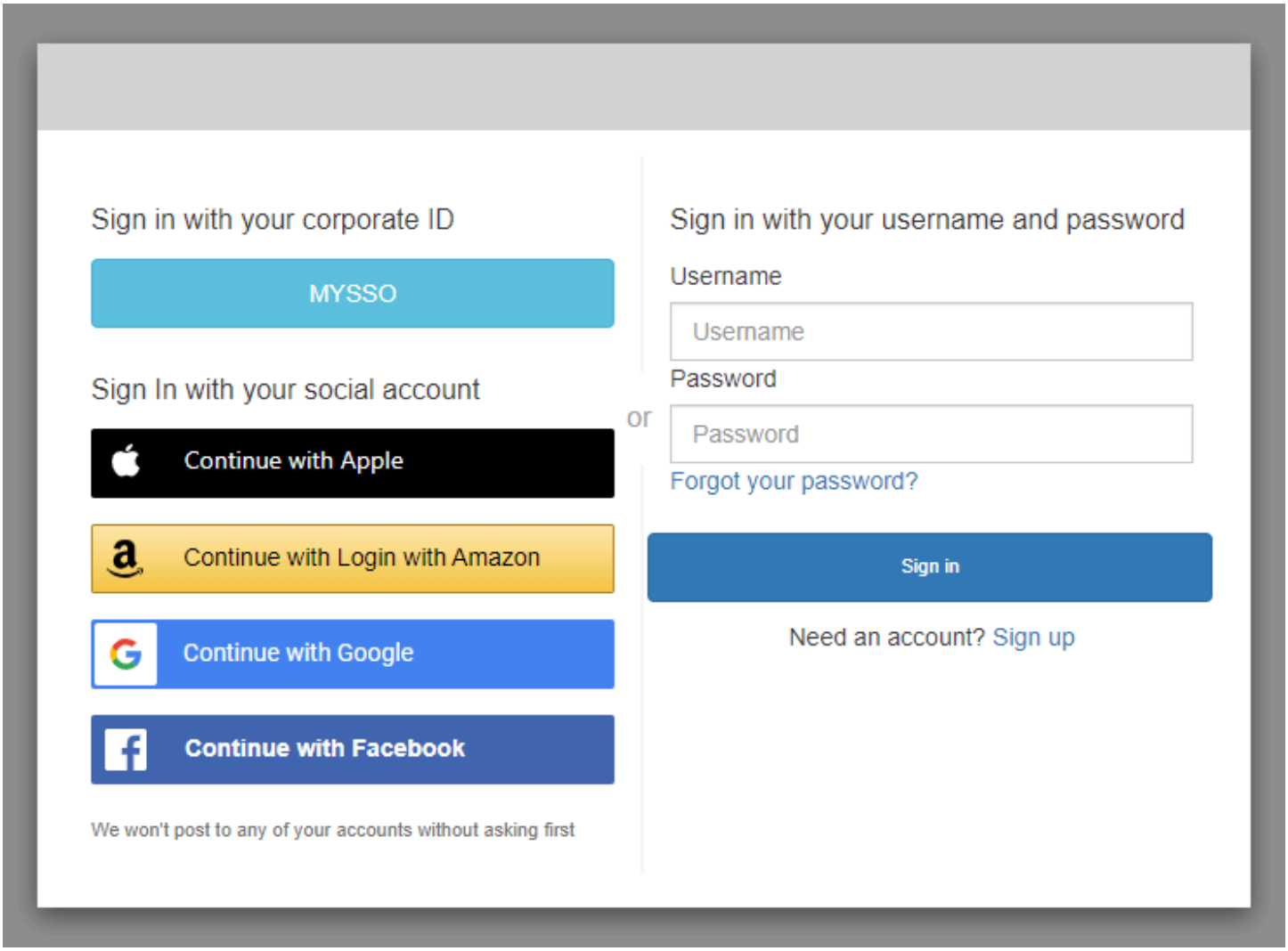
Argomenti

- Endpoint Login
- Endpoint Logout

Endpoint Login

L'endpoint login è un server di autenticazione e una destinazione di reindirizzamento da Endpoint Authorize. È il punto di ingresso all'interfaccia utente ospitata quando non si specifica un provider di identità. Quando generi un reindirizzamento all'endpoint di accesso, la pagina di accesso viene caricata e vengono presentate le opzioni di autenticazione configurate per il client all'utente.

> **Note**
>
> L'endpoint login è un componente dell'interfaccia utente ospitata. Nell'app, richiama la federazione e le pagine dell'interfaccia utente ospitate che effettuano il reindirizzamento all'endpoint login. L'accesso diretto da parte degli utenti all'endpoint di accesso non è una best practice.



GET /login

L'endpoint `/login` supporta solo `HTTPS GET` per la richiesta iniziale dell'utente. L'app richiama la pagina in un browser come Chrome o Firefox. Quando si esegue il reindirizzamento `/login` da aEndpoint Authorize, vengono trasmessi tutti i parametri forniti nella richiesta iniziale. L'endpoint di accesso supporta tutti i parametri di richiesta dell'endpoint Authorize. Puoi anche accedere direttamente all'endpoint di accesso. Come best practice, crea tutte le sessioni degli utenti in `/oauth2/authorize`.

Esempio: richiedere all'utente di accedere

In questo esempio viene visualizzata la schermata di accesso.

```
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
```

Esempio: risposta

Il server di autenticazione reindirizza all'app con lo stato e il codice di autorizzazione. IL server deve restituire il codice e lo stato nei parametri della stringa di query e non nel frammento.

```
HTTP/1.1 302 Found
                    Location: https://YOUR_APP/redirect_uri?
code=AUTHORIZATION_CODE&state=STATE
```

Richiesta di accesso avviata dall'utente

Dopo aver caricato l'endpoint `/login`, l'utente può inserire un nome utente e una password e scegliere Accedi. Quando lo fanno, generano una richiesta `HTTPS POST` con gli stessi parametri di richiesta di intestazione come richiesta `GET` e un corpo della richiesta con il nome utente, la password e l'impronta digitale del dispositivo.

Endpoint Logout

L'endpoint `/logout` è un endpoint di reindirizzamento. Disconnette l'utente e lo reindirizza a un URL di disconnessione autorizzato per il client dell'app o all'endpoint. `/login` I parametri disponibili in una

richiesta GET all'endpoint `/logout` sono personalizzati per i casi d'uso dell'interfaccia utente ospitata Amazon Cognito.

Per reindirizzare l'utente all'interfaccia utente ospitata per effettuare nuovamente l'accesso, aggiungi un parametro `redirect_uri`alla richiesta. Una richiesta `logout` con un parametro `redirect_uri` deve includere anche i parametri per la richiesta successiva a Endpoint Login, come `client_id`, `response_type` e `scope`.

L'endpoint di logout è un'applicazione web front-end per sessioni utente interattive con i tuoi clienti. L'app deve richiamare questo e altri endpoint dell'interfaccia utente ospitata nei browser degli utenti.

Per reindirizzare l'utente a una pagina preferita, aggiungi URL di disconnessione consentiti all'app del client. Nelle richieste degli utenti all'endpoint `logout`, aggiungi i parametri `logout_uri` e `client_id`. Se il valore di `logout_uri` è uno degli URL di disconnessione consentiti per il client dell'app, Amazon Cognito reindirizza gli utenti a tale URL.

Con il single logout (SLO) per SAML 2.0, Amazon IdPs Cognito reindirizza innanzitutto l'utente all'endpoint SLO definito nella configurazione IdP. Dopo che il tuo IdP ha reindirizzato l'utente a, Amazon `saml2/logout` Cognito risponde con un altro reindirizzamento alla o dalla tua richiesta. `redirect_uri` `logout_uri` Per ulteriori informazioni, consulta Flusso di disconnessione SAML.

L'endpoint di logout non disconnette gli utenti dall'OIDC o dai provider di identità social (). IdPs Per disconnettere gli utenti dalla sessione con un IdP esterno, indirizzali alla pagina di disconnessione di quel provider.

GET /logout

L'endpoint `/logout` supporta solo `HTTPS GET`. Il client del bacino d'utenza in genere invia questa richiesta tramite un browser di sistema. Il browser è in genere Custom Chrome Tab in Android o Safari View Control in iOS.

Parametri della richiesta

client_id

> L'ID client dell'app per l'app. Per ottenere l'ID del client di un'app, devi registrare l'app nel bacino d'utenza. Per ulteriori informazioni, consulta Client dell'app pool di utenti.
>
> Obbligatorio.

logout_uri

Reindirizza l'utente a una pagina di disconnessione personalizzata con un parametro logout_uri. Imposta il suo valore sull'URL di disconnessione del client dell'app a cui vuoi reindirizzare l'utente dopo la disconnessione. Utilizza logout_uri solo con il parametro client_id. Per ulteriori informazioni, consulta Client dell'app pool di utenti.

Puoi utilizzare il parametro logout_uri anche per reindirizzare l'utente alla pagina di accesso di un altro client di app. Imposta la pagina di accesso per l'altro client di app come Allowed callback URL (URL di callback consentito) nel tuo client di app. Nella richiesta all'endpoint `/logout`, imposta il valore del parametro logout_uri sulla pagina di accesso con codifica URL.

Amazon Cognito richiede un parametro logout_uri o redirect_uri nella richiesta all'endpoint `/logout`. Il parametro logout_uri reindirizza l'utente a un altro sito Web. Se nella richiesta all'endpoint `/logout` sono inclusi entrambi i parametri logout_uri e redirect_uri, Amazon Cognito utilizzerà esclusivamente il parametro logout_uri, sovrascrivendo il parametro redirect_uri.

redirect_uri

Reindirizza l'utente alla pagina di accesso per l'autenticazione con il parametro redirect_uri. Imposta il suo valore sull'URL di callback consentito del client dell'app dove vuoi reindirizzare l'utente quando questo ha nuovamente effettuato l'accesso. Utilizza i parametri client_id, scope, state e response_type che vuoi passare all'endpoint `/login`.

Amazon Cognito richiede un parametro logout_uri o redirect_uri nella richiesta all'endpoint `/logout`. Per reindirizzare l'utente all'*`/login`*endpoint per effettuare nuovamente l'autenticazione e passare i token all'app, aggiungi un parametro redirect_uri. Se nella richiesta all'endpoint sono inclusi entrambi i parametri logout_uri e redirect_uri, `/logout` Amazon Cognito sovrascrive il parametro redirect_uri ed elabora esclusivamente il parametro logout_uri.

response_type

La risposta OAuth 2.0 che desideri ricevere da Amazon Cognito dopo l'accesso dell'utente. `code` e `token` sono i valori validi per il parametro response_type.

Obbligatorio se si utilizza un parametro redirect_uri.

stato

Quando l'applicazione aggiunge un parametro di stato a una richiesta, Amazon Cognito ne restituisce il valore all'app quando l'`/oauth2/logout`endpoint reindirizza l'utente.

Aggiungi questo valore alle richieste di protezione contro attacchi CSRF.

Non è possibile impostare il valore di un parametro `state` su una stringa JSON con codifica URL. Per passare una stringa che corrisponda a questo formato in un `state` parametro, codifica la stringa in base64, quindi decodificala nell'applicazione.

Vivamente consigliato quando si utilizza un parametro redirect_uri.

scope

Gli ambiti OAuth 2.0 che desideri richiedere ad Amazon Cognito dopo averli disconnessi con un parametro redirect_uri. Amazon Cognito reindirizza l'utente all'endpoint `/login` con il parametro scopenella richiesta all'endpoint `/logout`.

Facoltativo quando si utilizza un parametro redirect_uri. Se non includi un parametro scope, Amazon Cognito reindirizza l'utente all'endpoint `/login` con un parametro scope. Quando Amazon Cognito reindirizza l'utente e compila automaticamente `scope`, il parametro include tutti gli ambiti autorizzati per il client di app.

Richieste di esempio

Esempio: disconnettersi e reindirizzare l'utente al client

Ad eccezione di `logout_uri` e`client_id`, tutti i possibili parametri di interrogazione per questo endpoint vengono passati a. Endpoint Authorize Quando le richieste includono `logout_uri` e `client_id`, Amazon Cognito reindirizza le sessioni utente all'URL nel valore di `logout_uri`, ignorando tutti gli altri parametri di richiesta. Questo URL deve essere un URL di disconnessione autorizzato per il client dell'app.

Di seguito è riportato un esempio di richiesta di disconnessione e reindirizzamento a `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
```

Esempio: disconnettersi e richiedere all'utente di accedere come altro utente

Quando le richieste omettono `logout_uri` ma forniscono in altro modo i parametri che costituiscono una richiesta ben formata all'endpoint di autorizzazione, Amazon Cognito reindirizza gli utenti all'accesso all'interfaccia utente ospitata. L'endpoint di disconnessione aggiunge i parametri della

richiesta originale alla destinazione di reindirizzamento. Il parametro `redirect_uri` in una richiesta all'endpoint di disconnessione non è un URL di disconnessione, ma un URL di accesso che desideri trasmettere all'endpoint di autorizzazione.

Di seguito è riportato un esempio di richiesta che disconnette un utente, reindirizza alla pagina di accesso e fornisce un codice di autorizzazione dopo l'accesso. `https://www.example.com`

```
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
```

Documentazione di riferimento degli endpoint di federazione OAuth 2.0, OpenID Connect e SAML 2.0

Amazon Cognito attiva gli endpoint in questa sezione quando si aggiunge un dominio al pool di utenti. Gli endpoint di federazione non sono elementi interattivi. Svolgono un ruolo di servizio per consentire alla tua app di comunicare con provider di identità OAuth 2.0, OIDC e SAML 2.0 di terze parti (). IdPs

Negli argomenti di questa guida vengono descritti diversi endpoint OAuth 2.0 e OIDC usati di frequente. Amazon Cognito crea i seguenti endpoint quando assegni un dominio al pool di utenti.

Endpoint di federazione del pool di utenti

URL dell'endpoint	Descrizione	Modalità di accesso
https://*dominio del pool di utenti*/oauth2/authorize	Reindirizza un utente all'interfaccia utente ospitata o all'accesso con il proprio IdP.	Richiamato nel browser del cliente per iniziare l'autenticazione dell'utente. Per informazioni, consulta Endpoint Authorize.
https://*dominio del pool di utenti*/oauth2/token	Restituisce i token in base a un codice di autorizzazione o richiesta di credenziali del client.	Richiesto dall'app per recuperare i token. Per informazioni, consulta Endpoint Token.

URL dell'endpoint	Descrizione	Modalità di accesso
https://*dominio del pool di utenti*/oauth2/userInfo	Restituisce gli attributi utente in base agli ambiti OAuth 2.0 e all'identità utente in un token di accesso.	Richiesto dall'app per recuperare il profilo utente. Per informazioni, consulta Endpoint UserInfo.
https://*dominio del pool di utenti*/oauth2/revoke	Revoca un token di aggiornamento e i token di accesso associati.	Richiesta dall'app di revocare un token. Per informazioni, consulta Endpoint Revoke.
https://cognito-idp.*Regione*.amazonaws.com/*ID pool di utenti*/.well-known/openid-configuration	Una directory dell'architettura OIDC del pool di utenti.	Richiesto dall'app per individuare i metadati dell'emittente del pool di utenti.
https://cognito-idp.*Regione*.amazonaws.com/*ID pool di utenti*/.well-known/jwks.json	Chiavi pubbliche che puoi utilizzare per convalidare i token di Amazon Cognito.	Richiesto dall'app per verificare i JWT.
https://*dominio pool di utenti*/oauth2/idpresponse	I gestori dell'identità digitale devono reindirizzare gli utenti a questo endpoint con un codice di autorizzazione. Amazon Cognito riscatta il codice per un token quando autentica l'utente federato.	Reindirizzato dall'accesso IdP OIDC come URL di callback del client IdP.
https://*dominio pool di utenti*/saml2/idpresponse	L'URL Assertion Consumer Response (ACS) per l'integrazione con i provider di identità SAML 2.0.	Reindirizzato da SAML 2.0 IdP come URL ACS o punto di origine per l'accesso avviato dall'IdP. [1]

URL dell'endpoint	Descrizione	Modalità di accesso
https://*Il dominio del* tuo pool di utenti /saml2/logout	L'URL Single Logout (SLO) per l'integrazione con i provider di identità SAML 2.0.	Reindirizzato da SAML 2.0 IdP come URL di accesso singolo (SLO). Accetta solo l'associazione POST.

[1] Per ulteriori informazioni sull'accesso SAML avviato da IdP, consulta. Utilizzo dell'accesso SAML avviato da IdP

Per ulteriori informazioni sugli standard OpenID Connect e OAuth, consultare OpenID Connect 1.0 e OAuth 2.0.

Argomenti

- Endpoint Authorize
- Endpoint Token
- Endpoint UserInfo
- Endpoint Revoke
- endpoint saml2/idpresponse

Endpoint Authorize

L'endpoint `/oauth2/authorize` è un endpoint di reindirizzamento che supporta due destinazioni di reindirizzamento. Se si include un `identity_provider` o `idp_identifier` nell'URL, viene eseguito il reindirizzamento invisibile all'utente alla pagina di accesso del gestore dell'identità digitale (IdP) specificato. In caso contrario, viene eseguito il reindirizzamento all'Endpoint Login con gli stessi parametri URL inclusi nella richiesta.

L'endpoint Autorizza reindirizza gli utenti all'interfaccia utente ospitata o alla pagina di accesso IdP. La destinazione di una sessione utente su questo endpoint è una pagina web con cui l'utente deve interagire direttamente nel proprio browser.

Per utilizzare l'endpoint di autorizzazione, richiama il browser dell'utente all'indirizzo `/oauth2/authorize` con parametri che forniscono al pool di utenti informazioni sui seguenti dettagli del pool stesso.

- Il client dell'app a cui vuoi accedere.

- L'URL di callback a cui si desidera essere reindirizzati.
- Gli ambiti OAuth 2.0 che desideri richiedere nel token di accesso dell'utente.
- Facoltativamente, il gestore dell'identità digitale (IdP) di terze parti che desideri utilizzare per accedere.

Puoi anche specificare i parametri `state` e `nonce` utilizzati da Amazon Cognito per convalidare le richieste in arrivo.

GET **`/oauth2/authorize`**

L'endpoint `/oauth2/authorize` supporta solo `HTTPS GET`. L'app in genere avvia questa richiesta nel browser dell'utente. È possibile effettuare richieste solo all'endpoint `/oauth2/authorize` tramite HTTPS.

È possibile trovare ulteriori informazioni sulla definizione dell'endpoint di autorizzazione nello standard OpenID Connect (OIDC) nella pagina relativa all'endpoint di autorizzazione.

Parametri della richiesta

`response_type`

(Obbligatorio) Il tipo di risposta. Deve essere `code` o `token`.

Una richiesta riuscita con `response_type` impostato su `code` restituisce una concessione del codice di autorizzazione. Una concessione del codice di autorizzazione è un parametro `code` che Amazon Cognito aggiunge all'URL di reindirizzamento. L'app scambia il codice con l'Endpoint Token per i token di accesso, ID e aggiornamento. Come best practice di sicurezza e per ricevere token di aggiornamento per gli utenti, usa una concessione del codice di autorizzazione nella tua app.

Una richiesta riuscita con `response_type` impostato su `token` restituisce una concessione implicita. Una concessione implicita è un token di ID e accesso che Amazon Cognito aggiunge all'URL di reindirizzamento. Una concessione implicita è meno sicura perché espone i token e le potenziali informazioni di identificazione agli utenti. Puoi disattivare il supporto per le concessioni implicite nella configurazione del client dell'app.

`client_id`

(Obbligatorio) L'ID del client dell'app.

Il valore del parametro `client_id` deve essere l'ID di un client dell'app nel pool di utenti in cui si effettua la richiesta. Il client dell'app deve supportare l'accesso da parte degli utenti locali di Amazon Cognito o di almeno un gestore dell'identità digitale (IdP) di terze parti.

`redirect_uri`

(Obbligatorio) L'URL a cui il server di autenticazione reindirizza il browser dopo che Amazon Cognito ha autorizzato l'utente.

L'URI (Uniform Resource Identifier) di reindirizzamento deve avere i seguenti attributi:

- Deve essere un URI assoluto.
- Deve essere preregistrato con un client.
- Non può includere un componente frammento.

Consulta OAuth 2.0 - Endpoint per il reindirizzamento.

Amazon Cognito richiede che l'URI di reindirizzamento utilizzi HTTPS, ad eccezione di `http://localhost`, che è possibile impostare come URL di callback a scopo di test.

Amazon Cognito supporta anche URL di callback come `myapp://example`.

`state`

(Facoltativo, consigliato) Quando l'app aggiunge un parametro di stato a una richiesta, Amazon Cognito ne restituisce il valore all'app quando l'`/oauth2/authorize`endpoint reindirizza l'utente.

Aggiungi questo valore alle richieste di protezione contro attacchi CSRF.

Non è possibile impostare il valore di un parametro `state` su una stringa JSON con codifica URL. Per passare una stringa che corrisponda a questo formato in un `state` parametro, codifica la stringa in base64, quindi decodificala nell'app.

`identity_provider`

(Facoltativo) Aggiungi questo parametro per ignorare l'interfaccia utente ospitata e reindirizzare l'utente a una pagina di accesso del provider. Il valore del parametro identity_provider è il nome del provider di identità come appare nel bacino d'utenza.

- Per i provider di social network, puoi utilizzare i valori identity_provider,, e. `Facebook` `Google` `LoginWithAmazon` `SignInWithApple`
- Per i pool di utenti di Amazon Cognito, usa il valore. `COGNITO`
- Per i provider di identità SAML 2.0 e OpenID Connect (OIDC) (IdPs), usa il nome che hai assegnato all'IdP nel tuo pool di utenti.

`idp_identifier`

(Facoltativo) Aggiungi questo parametro per reindirizzare a un provider con un nome alternativo per il nome identity_provider. Puoi inserire gli identificatori per SAML 2.0 e OIDC IdPs dalla scheda Esperienza di accesso della console Amazon Cognito.

`scope`

(Facoltativo) Può essere una combinazione di qualsiasi ambito riservato del sistema o ambito personalizzato associato a un client. Gli ambiti devono essere separati da spazi. Gli ambiti riservati al sistema sono `openid`, `email`, `phone`, `profile` e `aws.cognito.signin.user.admin`. Qualsiasi ambito utilizzato deve essere associato al client o verrà ignorato in fase di runtime.

Se il client non richiede alcun ambito, il server di autenticazione utilizza tutti gli ambiti associati al client.

Un token ID viene restituito solo viene richiesto l'ambito `openid`. Il token di accesso può essere utilizzato in relazione ai bacini d'utenza di Amazon Cognito solo se viene richiesto l'ambito `aws.cognito.signin.user.admin`. Gli ambiti `phone`, `email` e `profile` possono essere richiesti solo se viene richiesto anche l'ambito `openid`. Questi ambiti impongono le attestazioni da includere all'interno del token ID.

`code_challenge_method`

(Facoltativo) Il protocollo di hashing utilizzato per generare la sfida. Il PKCE RFC definisce due metodi, S256 e semplice; tuttavia, il server di autenticazione di Amazon Cognito supporta solo S256.

`code_challenge`

(Facoltativo) La sfida che hai generato da. `code_verifier`

Obbligatorio solo quando specifichi un parametro `code_challenge_method`.

`nonce`

(Facoltativo) Un valore casuale che puoi aggiungere alla richiesta. Il valore nonce fornito è incluso nel token ID emesso da Amazon Cognito. Per proteggersi da attacchi di tipo replay, l'app può analizzare la richiesta `nonce` nel token dell'ID e confrontarlo con quello generato. Per ulteriori informazioni sulla richiesta `nonce`, consulta la sezione relativa alla convalida del token dell'ID nella documentazione dello standard OpenID Connect.

Richieste di esempio con risposte positive

I seguenti esempi illustrano il formato delle richieste HTTP all'`/oauth2/authorize`endpoint.

Concessione codice autorizzazione

Questo è un esempio di richiesta per la concessione di un codice di autorizzazione.

Esempio: richiesta GET

La seguente richiesta avvia una sessione per recuperare un codice di autorizzazione che l'utente trasmette all'app nella `redirect_uri` destinazione. Questa sessione richiede gli ambiti per gli attributi utente e per l'accesso alle operazioni API self-service di 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
```

Esempio: risposta

Il server di autenticazione di Amazon Cognito viene reindirizzato all'app con lo stato e il codice di autorizzazione. Il codice di autorizzazione è valido per cinque minuti.

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Concessione del codice di autorizzazione con PKCE

Questo è un esempio di richiesta di concessione di un codice di autorizzazione con PKCE.

Esempio: richiesta GET

La richiesta seguente aggiunge un `code_challenge` parametro alla richiesta precedente. Per completare lo scambio di un codice con un token, è necessario includere il `code_verifier` parametro nella richiesta all'`/oauth2/token`endpoint.

```
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...
```

Esempio: risposta

Il server di autenticazione reindirizza all'applicazione con il codice e lo stato di autorizzazione. Il codice e lo stato devono essere restituiti nei parametri della stringa di query e non nel frammento:

```
HTTP/1.1 302 Found
Location: https://www.example.com?code=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111&state=abcdefg
```

Concessione del token senza ambito **`openid`**

Questa è una richiesta di esempio che genera una concessione implicita e restituisce JWT direttamente alla sessione dell'utente.

Esempio: richiesta GET

La seguente richiesta riguarda una concessione implicita dal server di autorizzazione. Il token di accesso di Amazon Cognito autorizza le operazioni API self-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
```

Esempio: risposta

Il server di autorizzazione di Amazon Cognito viene reindirizza all'app con il token di accesso. Poiché non è stato richiesto l'ambito del `openid`, Amazon Cognito non restituisce un token ID. Inoltre, Amazon Cognito non restituisce un token di aggiornamento in questo flusso. Amazon Cognito restituisce il token di accesso e lo stato nel frammento e non nella stringa di query:

```
HTTP/1.1 302 Found
```

```
Location: https://YOUR_APP/
redirect_uri#access_token=ACCESS_TOKEN&token_type=bearer&expires_in=3600&state=STATE
```

Concessione del token con ambito **openid**

Questo è un esempio di richiesta che genera una concessione implicita e restituisce JWT direttamente alla sessione dell'utente.

Esempio: richiesta GET

La seguente richiesta riguarda una concessione implicita dal server di autorizzazione. Il token di accesso di Amazon Cognito autorizza l'accesso agli attributi utente e alle operazioni API self-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
```

Esempio: risposta

Il server di autorizzazione reindirizza all'app con il token di accesso e il token ID (poiché l'`openid`ambito era incluso):

```
HTTP/1.1 302 Found
Location: https://
www.example.com#id_token=eyJra67890EXAMPLE&access_token=eyJra12345EXAMPLE&token_type=bearer&exp
```

Esempi di risposte negative

Amazon Cognito potrebbe rifiutare la tua richiesta. Le richieste negative sono accompagnate da un codice di errore HTTP e da una descrizione che puoi utilizzare per correggere i parametri della richiesta. Di seguito sono riportati alcuni esempi di risposte negative.

- Se `client_id` e `redirect_uri` sono validi, ma i parametri della richiesta non sono formattati correttamente, il server di autenticazione reindirizza l'errore al client `redirect_uri` e aggiunge un messaggio di errore in un parametro URL. Di seguito sono riportati alcuni esempi di formattazione errata.

- La richiesta non include un `response_type` parametro.
- La richiesta di autorizzazione ha fornito un `code_challenge` parametro, ma non un `code_challenge_method` parametro.
- Il valore del `code_challenge_method` parametro non lo è`S256`.

Di seguito è riportata la risposta a una richiesta di esempio con una formattazione errata.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_request
```

- Se il client richiede `code` o `token` accede`response_type`, ma non dispone dell'autorizzazione per queste richieste, il server di autorizzazione Amazon Cognito torna `unauthorized_client` a quello del client`redirect_uri`, come segue:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=unauthorized_client
```

- Se il client richiede un ambito non valido, sconosciuto o non corretto, il server di autorizzazione di Amazon Cognito restituisce `invalid_scope` al `redirect_uri` del client, come riportato di seguito:

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=invalid_scope
```

- Se si verifica un errore imprevisto nel server, il server di autenticazione torna `server_error` a quello del `redirect_uri` client. Poiché l'errore HTTP 500 non viene inviato al client, l'errore non viene visualizzato nel browser dell'utente. Il server di autorizzazione restituisce il seguente errore.

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?error=server_error
```

- Quando Amazon Cognito esegue l'autenticazione tramite federazione a terze parti, Amazon IdPs Cognito potrebbe riscontrare problemi di connessione, come i seguenti:
  - Se si verifica un timeout di connessione durante la richiesta del token dal provider di identità, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Timeout+occurred+in+calling+IdP+token
    +endpoint
    ```

  - Se si verifica un timeout di connessione durante la chiamata all'`jwks_uri`endpoint per la convalida del token ID, il server di autenticazione reindirizza con un errore al client nel modo seguente: `redirect_uri`

```
HTTP 1.1 302 Found Location: https://client_redirect_uri?
error=invalid_request&error_description=error_description=Timeout+in+calling+jwks
+uri
```

- Durante l'autenticazione mediante federazione a terze parti IdPs, i provider possono restituire risposte di errore. Ciò può essere dovuto a errori di configurazione o ad altri motivi, come i seguenti:
  - Se viene ricevuta una risposta di errore da altri provider, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=[IdP name]+Error+-+[status code]+error
     getting token
    ```

  - Se viene ricevuta una risposta di errore da Google, il server di autenticazione reindirizza l'errore al `redirect_uri` del client come segue:

    ```
    HTTP 1.1 302 Found Location: https://client_redirect_uri?
    error=invalid_request&error_description=Google+Error+-+[status code]+[Google-
    provided error code]
    ```

- Quando Amazon Cognito rileva un'eccezione di comunicazione quando si connette a un IdP esterno, il server di autenticazione reindirizza con un errore al client con uno dei seguenti messaggi: `redirect_uri`
  - ```
    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
    ```

Endpoint Token

L'endpoint del token OAuth 2.0 su `/oauth2/token` emette token web JSON (JWT).

Il server di autorizzazione OAuth 2.0 del tuo pool di utenti emette token web JSON (JWT) dall'endpoint del token ai seguenti tipi di sessioni:

1. Utenti che hanno completato una richiesta di concessione del codice di autorizzazione. Il riscatto riuscito di un codice restituisce i token ID, di accesso e di aggiornamento.
2. Sessioni M achine-to-machine (M2M) che hanno completato una concessione di credenziali client. L'autorizzazione riuscita con il segreto del client restituisce un token di accesso.
3. Utenti che hanno precedentemente effettuato l'accesso e ricevuto token di aggiornamento. L'autenticazione con token di aggiornamento restituisce nuovi ID e token di accesso.

> **Note**
>
> Gli utenti che accedono con un codice di autorizzazione concesso nell'interfaccia utente ospitata o tramite la federazione possono sempre aggiornare i propri token dall'endpoint del token. Gli utenti che accedono con le operazioni API `InitiateAuth` e `AdminInitiateAuth` possono aggiornare i propri token con l'endpoint token quando i dispositivi ricordati non sono attivi nel pool di utenti. Se i dispositivi ricordati sono attivi, aggiorna i token con le richieste `AuthFlow` of `REFRESH_TOKEN_AUTH` in o API. `InitiateAuth` `AdminInitiateAuth`

L'endpoint token diventa disponibile pubblicamente quando si aggiunge un dominio al pool di utenti. Accetta le richieste POST HTTP. Per la sicurezza delle applicazioni, utilizzate PKCE con gli eventi di accesso al codice di autorizzazione. PKCE verifica che l'utente che trasmette un codice di autorizzazione sia lo stesso utente che ha effettuato l'autenticazione. Per ulteriori informazioni su PKCE, vedere IETF RFC 7636.

Puoi saperne di più sui client dell'app del pool di utenti e sui relativi tipi di concessione, segreti dei client, ambiti autorizzati e ID client all'indirizzo Client dell'app pool di utenti. Puoi saperne di più sull'autorizzazione M2M, sulla concessione delle credenziali dei client e sull'autorizzazione con ambiti dei token di accesso all'indirizzo. Autorizzazione Scopes, M2M e API con server di risorse

Per recuperare informazioni su un utente dal suo token di accesso, passale alla tua richiesta Endpoint UserInfo o a una richiesta API. GetUser

POST /oauth2/token

L'endpoint `/oauth2/token` supporta solo `HTTPS POST`. L'app effettua direttamente le richieste a questo endpoint e non tramite il browser.

L'endpoint token supporta l'autenticazione `client_secret_basic` e `client_secret_post`. Per ulteriori informazioni sulla specifica OpenID Connect, vedere Client Authentication. Per ulteriori informazioni sull'endpoint Token della specifica OpenID Connect, consulta Endpoint Token.

Parametri della richiesta nell'intestazione

`Authorization`

Se il client ha emesso un segreto, dovrà trasmettere i suoi `client_id` e `client_secret` nell'intestazione dell'autorizzazione come autorizzazione HTTP `client_secret_basic`. Puoi inoltre includere `client_id` e `client_secret` nel corpo della richiesta come autorizzazione `client_secret_post`.

La stringa di intestazione di autorizzazione è Base `Base64Encode(client_id:client_secret)`. L'esempio seguente è un'intestazione di autorizzazione per il client dell'app `djc98u3jiedmi283eu928` con client secret`abcdef01234567890`, che utilizza la versione della stringa con codifica Base64: `djc98u3jiedmi283eu928:abcdef01234567890`

```
Authorization: Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw
```

`Content-Type`

Imposta il valore di questo parametro su `'application/x-www-form-urlencoded'`.

Parametri della richiesta nel corpo

`grant_type`

(Obbligatorio) Il tipo di concessione OIDC che desideri richiedere.

Deve essere `authorization_code` o `refresh_token` o `client_credentials`. È possibile richiedere un token di accesso per un ambito personalizzato dall'endpoint del token alle seguenti condizioni:

- Hai abilitato l'ambito richiesto nella configurazione del client dell'app.
- Hai configurato il client dell'app con un client secret.
- Abilita la concessione delle credenziali del cliente nel client dell'app.

client_id

(Facoltativo) L'ID di un client dell'app nel tuo pool di utenti. Specificate lo stesso client dell'app che ha autenticato l'utente.

È necessario fornire questo parametro se il client è pubblico e non dispone di un segreto o è `client_secret_post` autorizzato. `client_secret`

client_secret

(Facoltativo) Il segreto del client dell'app che ha autenticato l'utente. Obbligatorio se il client app ha un segreto del client e non è stata inviata una intestazione `Authorization`.

scope

(Facoltativo) Può essere una combinazione di qualsiasi ambito personalizzato associato a un client di app. Qualsiasi ambito richiesto deve essere attivato per il client dell'app. In caso contrario, Amazon Cognito lo ignorerà. Se il client non richiede alcun ambito, il server di autenticazione assegna tutti gli ambiti personalizzati che hai autorizzato nella configurazione del client dell'app.

Utilizzato solo se il valore `grant_type` è `client_credentials`.

redirect_uri

(Facoltativo) Deve essere `redirect_uri` lo stesso usato per entrare. `authorization_code` `/oauth2/authorize`

È necessario fornire questo parametro se lo `grant_type` è`authorization_code`.

refresh_token

(Facoltativo) Per generare nuovi token di accesso e ID per la sessione di un utente, imposta il valore di un `refresh_token` parametro nella `/oauth2/token` richiesta su un token di aggiornamento emesso in precedenza dallo stesso client dell'app.

code

(Facoltativo) Il codice di autorizzazione derivante dalla concessione di un codice di autorizzazione. È necessario fornire questo parametro se la richiesta di autorizzazione includeva un `grant_type` di`authorization_code`.

code_verifier

(Facoltativo) Il valore arbitrario utilizzato per calcolare il valore contenuto `code_challenge` in una richiesta di concessione del codice di autorizzazione con PKCE.

Richieste di esempio con risposte positive

Sostituzione di un codice di autorizzazione per i token

Esempio: richiesta POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token&
                            Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                            grant_type=authorization_code&
                            client_id=1example23456789&
                            code=AUTHORIZATION_CODE&
                            redirect_uri=com.myclientapp://myclient/redirect
```

Esempio: risposta

```
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**
>
> L'endpoint del token restituisce `refresh_token` solo quando `grant_type` è `authorization_code`.

Sostituzione delle credenziali del client per un token di accesso: segreto del client nell'intestazione di autorizzazione

Esempio: richiesta POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
```

```
                            Content-Type='application/x-www-form-urlencoded'&

Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                            grant_type=client_credentials&
                            client_id=1example23456789&

scope=resourceServerIdentifier1/scope1 resourceServerIdentifier2/scope2
```

Esempio: risposta

```
HTTP/1.1 200 OK
                            Content-Type: application/json

                            {
                            "access_token":"eyJra1example",
                            "token_type":"Bearer",
                            "expires_in":3600
                            }
```

Sostituzione delle credenziali del client per un token di accesso: segreto del client nel corpo della richiesta

Esempio: richiesta 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
```

Esempio: risposta

```
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"
}
```

Sostituzione di una concessione del codice di autorizzazione con PKCE per i token

Esempio: richiesta POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token
                                Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                                grant_type=authorization_code&
                                client_id=1example23456789&
                                code=AUTHORIZATION_CODE&
                                code_verifier=CODE_VERIFIER&
                                redirect_uri=com.myclientapp://myclient/redirect
```

Esempio: risposta

```
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**
>
> L'endpoint del token restituisce `refresh_token` solo quando `grant_type` è `authorization_code`.

Sostituzione di un token di aggiornamento per i token

Esempio: richiesta POST

```
POST https://mydomain.auth.us-east-1.amazoncognito.com/oauth2/token >
                          Content-Type='application/x-www-form-urlencoded'&

 Authorization=Basic ZGpjOTh1M2ppZWRtaTI4M2V1OTI4OmFiY2RlZjAxMjM0NTY3ODkw

                          grant_type=refresh_token&
                          client_id=1example23456789&
                          refresh_token=eyJj3example
```

Esempio: risposta

```
HTTP/1.1 200 OK
                          Content-Type: application/json

                          {
                          "access_token":"eyJra1example",
                          "id_token":"eyJra2example",
                          "token_type":"Bearer",
                          "expires_in":3600
                                }
```

> **Note**
>
> L'endpoint del token restituisce `refresh_token` solo quando `grant_type` è `authorization_code`.

Esempi di risposte negative

Esempio: risposta all'errore

```
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`

Nella richiesta manca un parametro obbligatorio, include un valore di parametro non supportato (diverso da `unsupported_grant_type`) oppure è comunque non corretto. Ad esempio, `grant_type` è `refresh_token` ma `refresh_token` non è incluso.

`invalid_client`

L'autenticazione del client non è riuscita. Ad esempio, quando il client include `client_id` e `client_secret` nell'intestazione di autorizzazione, ma non esiste un client con valori `client_id` e `client_secret`.

`invalid_grant`

Il token di aggiornamento è stato revocato.

Il codice di autorizzazione è stato utilizzato già o non esiste.

Il client dell'app non ha accesso in lettura a tutti gli attributi nell'ambito richiesto. Ad esempio, l'app richiede l'ambito `email` e il client dell'app può leggere l'attributo `email`, ma non `email_verified`.

`unauthorized_client`

Il client non ha le autorizzazioni necessarie per il flusso di concessione del codice o per l'aggiornamento dei token.

`unsupported_grant_type`

Restituito se `grant_type` è diverso da `authorization_code`, `refresh_token` o `client_credentials`.

Endpoint UserInfo

L'endpoint `userInfo` è un endpoint userInfo OpenID Connect (OIDC). Risponde con gli attributi utente quando i fornitori di servizi presentano i token di accesso rilasciati da Endpoint Token. Gli ambiti nel token di accesso dell'utente definiscono gli attributi utente restituiti dall'endpoint userInfo nella sua risposta. L'ambito `openid` deve essere una delle richieste del token di accesso.

Amazon Cognito emette token di accesso in risposta a richieste API dei pool di utenti come InitiateAuth. Poiché non contengono alcun ambito, l'endpoint userInfo non accetta questi token di accesso. È invece necessario presentare i token di accesso dall'endpoint del token.

Il gestore dell'identità digitale (IdP) di terze parti OAuth 2.0 ospita anche un endpoint userInfo. Quando l'utente si autentica con quell'IdP, Amazon Cognito scambia silenziosamente un codice di autorizzazione con l'endpoint IdP. `token` Il tuo pool di utenti passa il token di accesso IdP per autorizzare il recupero delle informazioni utente dall'endpoint IdP. `userInfo`

GET /oauth2/userInfo

L'app effettua le richieste direttamente a questo endpoint e non tramite il browser.

Per ulteriori informazioni, consulta Endpoint UserInfo nelle specifiche di OpenID Connect (OIDC).

Argomenti

- Parametri della richiesta nell'intestazione
- Esempio: richiesta
- Esempio: risposta positiva
- Esempio di risposte negative

Parametri della richiesta nell'intestazione

`Authorization: Bearer` ***`<access_token>`***

Passa il token di accesso nel campo dell'intestazione dell'autorizzazione.

Obbligatorio.

Esempio: richiesta

```
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
```

Esempio: risposta positiva

```
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"
}
```

Per un elenco di richieste OIDC, vedi la sezione relativa alle richieste standard. Attualmente, Amazon Cognito restituisce i valori per `email_verified` e `phone_number_verified` come stringhe.

Esempio di risposte negative

Esempio: richiesta errata

```
HTTP/1.1 400 Bad Request
```

```
WWW-Authenticate: error="invalid_request",
error_description="Bad OAuth2 request at UserInfo Endpoint"
```

`invalid_request`

Nella richiesta manca un parametro obbligatorio, include un valore di parametro non supportato o è altrimenti non valida.

Esempio: token errato

```
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`

Il token di accesso è scaduto, revocato, non valido o non è valido.

Endpoint Revoke

L'`oauth2/revoke`endpoint/revoca il token di accesso di un utente che Amazon Cognito aveva inizialmente emesso con il token di aggiornamento fornito. Questo endpoint revoca inoltre tutti i token di accesso e di identità successivi dallo stesso token di aggiornamento. Dopo che l'endpoint revoca i token, non puoi utilizzare i token di accesso revocati per accedere alle API autenticate con i token Amazon Cognito.

POST /oauth2/revoke

L'endpoint `/oauth2/revoke` supporta solo `HTTPS POST`. Il client del bacino d'utenza effettua direttamente le richieste a questo endpoint e non tramite il browser di sistema.

Parametri della richiesta nell'intestazione

`Authorization`

Se il client dell'app dispone di un client secret, l'applicazione deve trasmetterlo `client_id` e inserirlo `client_secret` nell'intestazione di autorizzazione tramite l'autorizzazione HTTP di base. Il segreto è Basic `Base64Encode(client_id:client_secret)`.

`Content-Type`

Deve sempre essere `'application/x-www-form-urlencoded'`.

Parametri della richiesta nel corpo

`token`

(Obbligatorio) Il token di aggiornamento che il client desidera revocare. La richiesta revoca anche tutti i token di accesso emessi da Amazon Cognito con questo token di aggiornamento.

Obbligatorio.

`client_id`

(Facoltativo) L'ID client dell'app per il token che desideri revocare.

Obbligatorio se il client è pubblico e non dispone di un segreto.

Esempi di richiesta di revoca

Esempio 1: revoca di un token per un client dell'app senza un segreto del 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
```

Esempio 2: revoca di un token per un client dell'app con un segreto del 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
```

Risposta di errore di revoca

Una risposta riuscita contiene un corpo vuoto. La risposta di errore è un oggetto JSON con un campo `error` e possibilmente un campo `error_description`.

Errori di endpoint

- Se il token non è presente nella richiesta o se la funzionalità è disabilitata per il client dell'app, viene restituito un codice di errore HTTP 400 e `invalid_request`.
- Se il token inviato da Amazon Cognito nella richiesta di revoca non è un token di aggiornamento, riceverai un codice di errore HTTP 400 e `unsupported_token_type`.
- Se le credenziali client non sono valide, riceverai un codice di errore HTTP 401 e `invalid_client`.
- Se il token è stato revocato o se il client ha inviato un token non valido, viene restituito un codice di errore HTTP 200 OK.

endpoint saml2/idpresponse

`/saml2/idpresponse`riceve asserzioni SAML. Nell'accesso service-provider-initiated (avviato da SP), il tuo provider di identità SAML 2.0 (IdP) reindirizza l'utente a questo endpoint con la sua risposta SAML. Nell'accesso avviato da SP, l'applicazione non interagisce con questo endpoint. Configura il tuo IdP con il percorso del tuo `saml2/idpresponse` URL As the Assertion Consumer Service (ACS). Per ulteriori informazioni sull'avvio della sessione, consulta. Avvio della sessione SAML nei bacini d'utenza di Amazon Cognito

Nell'accesso avviato da IdP, gli utenti possono accedere con il tuo IdP tramite la tua procedura e inviare un'asserzione SAML nel corpo di una richiesta tramite HTTPS. `HTTP POST` Il corpo della `POST` richiesta deve essere un parametro e un parametro. `SAMLResponse` `Relaystate` Per ulteriori informazioni, consulta Utilizzo dell'accesso SAML avviato da IdP.

POST **`/saml2/idpresponse`**

Per utilizzare l'`/saml2/idpresponse`endpoint in un accesso avviato da IdP, genera una richiesta POST con parametri che forniscano al tuo pool di utenti informazioni sulla sessione dell'utente.

- Il client dell'app a cui vogliono accedere.
- L'URL di callback a cui vogliono finire.
- Gli ambiti OAuth 2.0 che vogliono richiedere nel token di accesso dell'utente.

- L'IdP che ha avviato la richiesta di accesso.

Parametri del corpo della richiesta avviati da IDP

Risposta SAML

Un'asserzione SAML con codifica Base64 di un IdP associata a un client di app valido e a una configurazione IdP nel tuo pool di utenti.

RelayState

Un `RelayState` parametro contiene i parametri di richiesta che altrimenti passeresti all'endpoint. `oauth2/authorize` Per informazioni dettagliate su questi parametri, vedereEndpoint Authorize.

response_type

Il tipo di concessione OAuth 2.0.

client_id

L'ID del client dell'applicazione.

redirect_uri

L'URL a cui il server di autenticazione reindirizza il browser una volta che Amazon Cognito concede l'autorizzazione all'utente.

identity_provider

Il nome del provider di identità a cui desideri reindirizzare l'utente.

idp_identifier

L'identificatore del provider di identità a cui desideri reindirizzare l'utente.

scope

Gli ambiti OAuth 2.0 che desideri che l'utente richieda al server di autorizzazione.

Richieste di esempio con risposte positive

Esempio: richiesta POST

La seguente richiesta riguarda la concessione di un codice di autorizzazione per un utente da IdP `MySAMLIdP` nel client dell'app. `1example23456789` L'utente reindirizza a `https://`

`www.example.com` con il proprio codice di autorizzazione, che può essere scambiato con token che includono un token di accesso con ambiti OAuth 2.0, e. `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
```

Esempio: risposta

Di seguito è riportata la risposta alla richiesta precedente.

```
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]
```

Concessioni OAuth 2.0

Il server di autorizzazione OAuth 2.0 del pool di utenti Amazon Cognito emette token in risposta a tre tipi di OAuth 2.0 di concessioni di autorizzazione. Puoi impostare i tipi di concessione supportati per ogni client dell'app nel pool di utenti. Non puoi abilitare concessioni delle credenziali del client nello stesso client dell'app come concessioni implicite o del codice di autorizzazione. Le richieste per concessioni implicite e del codice di autorizzazione iniziano in in Endpoint Authorize e le richieste per concessioni delle credenziali dei clienti iniziano in Endpoint Token.

Concessione codice autorizzazione

> In risposta alla richiesta di autenticazione riuscita, il server di autorizzazione aggiunge un codice di autorizzazione in un parametro `code` all'URL di callback. Occorre quindi scambiare il codice per token ID, di accesso e di aggiornamento con Endpoint Token. Per richiedere una concessione del codice di autorizzazione, imposta `response_type` su `code` nella richiesta. Per una richiesta di esempio, consulta Concessione codice autorizzazione.

La concessione del codice di autorizzazione è la forma più sicura di concessione di autorizzazione. Non mostra il contenuto dei token direttamente agli utenti. L'app è invece responsabile del recupero e dell'archiviazione sicura dei token dell'utente. In Amazon Cognito, una concessione del codice di autorizzazione è l'unico modo per ottenere tutti e tre i tipi di token (ID, accesso e aggiornamento) dal server di autorizzazione. Puoi anche ottenere tutti e tre i tipi di token dall'autenticazione tramite l'API dei pool di utenti di Amazon Cognito, ma l'API non emette token di accesso con ambiti diversi da `aws.cognito.signin.user.admin`.

Implicit grant (Concessione implicita)

In risposta alla richiesta di autenticazione riuscita, il server di autorizzazione aggiunge un token di accesso in un parametro `access_token` e un token ID in un parametro `id_token`, all'URL di callback. Una concessione implicita non richiede alcuna interazione aggiuntiva con Endpoint Token. Per richiedere una concessione implicita, imposta `response_type` su `token` nella richiesta. La concessione implicita genera solo un ID e un token di accesso. Per una richiesta di esempio, consulta Concessione del token senza ambito `openid`.

La concessione implicita è una concessione di autorizzazione legacy. A differenza della concessione del codice di autorizzazione, gli utenti possono intercettare e ispezionare i token. Per impedire la distribuzione dei token tramite concessione implicita, configura il client dell'app in modo che supporti solo la concessione del codice di autorizzazione.

Client credentials (Credenziali del client)

Le credenziali del client sono concesse solo in base all'autorizzazione per l'accesso. machine-to-machine Per ricevere una concessione delle credenziali del client, ignora il Endpoint Authorize e genera una richiesta direttamente a Endpoint Token. Il client dell'app deve disporre di un segreto del client e supportare solo le concessioni delle credenziali del cliente. In risposta alla richiesta riuscita, il server di autorizzazione restituisce un token di accesso.

Il token di accesso proveniente dalla concessione delle credenziali del client è un meccanismo di autorizzazione che contiene ambiti OAuth 2.0. In genere, il token contiene attestazioni di ambito personalizzate che autorizzano operazioni HTTP alle API ad accesso protetto. Per ulteriori informazioni, consulta Autorizzazione Scopes, M2M e API con server di risorse.

Le concessioni di credenziali al cliente aggiungono costi alla fattura. AWS Per ulteriori informazioni, consultare Prezzi di Amazon Cognito.

Utilizzo di PKCE nelle concessioni di codici di autorizzazione con pool di utenti Amazon Cognito

Amazon Cognito supporta l'autenticazione Proof Key for Code Exchange (PKCE) nelle concessioni di codici di autorizzazione. PKCE è un'estensione della concessione del codice di autorizzazione OAuth 2.0 per i clienti pubblici. PKCE protegge dal riscatto dei codici di autorizzazione intercettati.

In che modo Amazon Cognito utilizza PKCE

Per avviare l'autenticazione con PKCE, l'applicazione deve generare un valore di stringa univoco. Questa stringa è il verificatore del codice, un valore segreto che Amazon Cognito utilizza per confrontare il client che richiede la concessione di autorizzazione iniziale con il client che scambia il codice di autorizzazione in token.

L'app deve applicare un hash SHA256 alla stringa del verificatore del codice e codificare il risultato in base64. Passa la stringa con hash al parametro as nel corpo della richiesta. Endpoint Authorize `code_challenge` Quando l'app scambia il codice di autorizzazione in cambio di token, deve includere la stringa del verificatore del codice in testo semplice come `code_verifier` parametro nel corpo della richiesta a. Endpoint Token Amazon Cognito esegue la stessa hash-and-encode operazione sul verificatore di codice. Amazon Cognito restituisce ID, accesso e token di aggiornamento solo se determina che il verificatore del codice genera la stessa richiesta di codice ricevuta nella richiesta di autorizzazione.

Per implementare Authorization Grant Flow con PKCE

1. Apri la console Amazon Cognito. Se richiesto, inserisci le tue AWS credenziali.
2. Scegli User Pools (Pool di utenti).
3. Scegli un bacino d'utenza esistente dall'elenco o creane uno nuovo. Se crei un pool di utenti, durante la procedura guidata ti verrà richiesto di configurare un client per l'app e configurare l'interfaccia utente ospitata.

    a. Se crei un nuovo pool di utenti, configura un client di app e configura l'interfaccia utente ospitata durante la configurazione guidata.

    b. Se configuri un pool di utenti esistente, aggiungi un dominio e un client di app pubblico, se non l'hai già fatto.
4. Genera una stringa alfanumerica casuale, in genere un identificatore univoco universale (UUID), per creare una sfida di codice per il PKCE. Questa stringa è il valore del `code_verifier` parametro che verrà inviato nella richiesta a. Endpoint Token

5. Effettua l'hash della `code_verifier` stringa con l'algoritmo SHA256. Codifica il risultato dell'operazione di hashing in base64. Questa stringa è il valore del `code_challenge` parametro che verrà inviato nella richiesta a. Endpoint Authorize

   L'Pythonesempio seguente genera un `code_verifier` e calcola: `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}")
   ```

   Di seguito è riportato un esempio di output dello Python script:

   ```
   code challenge: Eh0mg-OZv7BAyo-tdv_vYamx1boOYDulDklyXoMDtLg
   code verifier: 9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
   JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
   ```

6. Completa l'accesso all'interfaccia utente ospitata con una richiesta di concessione del codice di autorizzazione con PKCE. Di seguito è riportato un esempio di URL:

   ```
   https://mydomain.us-east-1.amazoncognito.com/oauth2/authorize?
   response_type=code&client_id=1example23456789&redirect_uri=https://
   www.example.com&code_challenge=Eh0mg-OZv7BAyo-
   tdv_vYamx1boOYDulDklyXoMDtLg&code_challenge_method=S256
   ```

7. Raccogli l'autorizzazione `code` e riscattala in token con l'endpoint del token. Di seguito è riportato un esempio di richiesta:

```
POST /oauth2/token HTTP/1.1
Host: mydomain.us-east-1.amazoncognito.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 296

redirect_uri=https%3A%2F%2Fwww.example.com&
client_id=1example23456789&
code=7378f445-c87f-400c-855e-0297d072ff03&
grant_type=authorization_code&
code_verifier=9D-aW_iygXrgQcWJd0y0tNVMPSXSChIc2xceDhvYVdGLCBk-
JWFTmBNjvKSdOrjTTYazOFbUmrFERrjWx6oKtK2b6z_x4_gHBDlr4K1mRFGyE8yA-05-_v7Dxf3EIYJH
```

8. Rivedi la risposta. Conterrà ID, accesso e token di aggiornamento. Per ulteriori informazioni sull'utilizzo dei token del pool di utenti di Amazon Cognito, consulta. Utilizzo di token con bacini d'utenza

Risposte agli errori di federazione e dell'interfaccia utente ospitata

Un processo di accesso nell'interfaccia utente ospitata o l'accesso federato potrebbero restituire un errore. Di seguito sono riportate alcune condizioni che possono determinare un errore di autenticazione.

- Un utente esegue un'operazione che il pool di utenti non è in grado di eseguire.
- Un trigger Lambda non risponde con la sintassi prevista.
- Il gestore dell'identità digitale restituisce un errore.
- Amazon Cognito non è riuscito a convalidare le informazioni sugli attributi fornite dall'utente.
- L'IdP non ha inviato richieste che corrispondono agli attributi richiesti.

Quando Amazon Cognito rileva un errore, lo comunica in uno dei seguenti modi.

1. Amazon Cognito invia un URL di reindirizzamento con l'errore nei parametri della richiesta.
2. Amazon Cognito visualizza un errore nell'interfaccia utente ospitata.

Gli errori aggiunti da Amazon Cognito ai parametri della richiesta hanno il seguente formato.

```
https://<Callback URL>/?error_description=error+description&error=error+name
```

Quando fornisci assistenza agli utenti per inviare informazioni sugli errori quando non è possibile eseguire un'operazione, richiedi che acquisiscano l'URL e il testo o uno screenshot della pagina.

> **Note**
>
> Le descrizioni degli errori di Amazon Cognito non sono stringhe fisse e non devi usare una logica basata su un modello o un formato fisso.

Messaggi di errore del provider di identità OIDC e social

Il provider di identità potrebbe restituire un errore. Quando un IdP OIDC o OAuth 2.0 restituisce un errore conforme agli standard, Amazon Cognito reindirizza l'utente all'URL di callback e aggiunge la risposta di errore del provider ai parametri della richiesta di errore. Amazon Cognito aggiunge il nome del provider e il codice di errore HTTP alle stringhe di errore esistenti.

L'URL seguente è un reindirizzamento di esempio da un IdP che ha restituito un errore ad 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
```

Poiché Amazon Cognito restituisce solo ciò che riceve da un provider, l'utente potrebbe visualizzare un sottoinsieme di queste informazioni.

Quando l'utente rileva un problema con l'accesso iniziale tramite l'IdP, questo invia eventuali messaggi di errore direttamente all'utente. Amazon Cognito inoltra un messaggio di errore all'utente quando genera una richiesta all'IdP per convalidare la sessione dell'utente. Amazon Cognito inoltra i messaggi di errore IdP OAuth e OIDC dai seguenti endpoint.

`/token`

Amazon Cognito scambia un codice di autorizzazione IdP per un token di accesso.

`/.well-known/openid-configuration`

Amazon Cognito rileva il percorso agli endpoint dell'emittente.

`/.well-known/jwks.json`

Per verificare i token web JSON dell'utente, Amazon Cognito rileva le JSON Web Key (JWK) utilizzate dall'IdP per firmare i token.

Poiché Amazon Cognito non avvia sessioni in uscita verso i provider SAML 2.0 che potrebbero restituire errori HTTP, gli errori degli utenti durante una sessione con un IdP SAML 2.0 non includono questa forma di messaggio di errore del provider.

Documentazione di riferimento API dei pool di utenti di Amazon Cognito

Con i pool di utenti di Amazon Cognito, è possibile consentire agli utenti di registrarsi e accedere alle app Web e per dispositivi mobili. È possibile modificare le password per gli utenti autenticati e avviare flussi di password dimenticata per gli utenti non autenticati. Per ulteriori informazioni, consultare Flusso di autenticazione del bacino d'utenza e Utilizzo di token con bacini d'utenza.

L'API dei pool di utenti di Amazon Cognito include operazioni per visualizzare e modificare i pool di utenti e gli utenti e per eseguire l'autenticazione e l'autorizzazione utente. Per una descrizione delle classi di operazioni API che si combinano nell'API dei pool di utenti di Amazon Cognito, consulta Utilizzo dell'API dei pool di utenti Amazon Cognito e degli endpoint del pool di utenti.

Per un elenco dettagliato delle operazioni e della sintassi delle API dei pool di utenti di Amazon Cognito, consulta Documentazione di riferimento dell'API dei pool di utenti di Amazon Cognito. Ogni pagina nella Documentazione di riferimento dell'API dei pool di utenti di Amazon Cognito si collega a materiale di riferimento con sintassi ed esempi per un'ampia gamma di SDK AWS.

Documentazione di riferimento dell'API dei pool di identità di Amazon Cognito (identità federate)

Con un pool di identità di Amazon Cognito, gli utenti delle app Web e per dispositivi mobili possono ottenere credenziali AWS con privilegi temporanei e limitati che consentono loro di accedere ad altri servizi AWS.

Per il materiale di riferimento completo sull'API dei pool di identità (identità federate), consulta Riferimento API di Amazon Cognito.

Documentazione di riferimento dell'API di Amazon Cognito sync

Amazon Cognito Sync è un servizio AWS e una libreria client che consente la sincronizzazione tra più dispositivi di dati dell'utente relativi all'applicazione.

Per ulteriori informazioni sul riferimento alle API di Amazon Cognito Sync, consulta la Documentazione di riferimento delle API di Amazon Cognito Sync.

Cronologia dei documenti per Amazon Cognito

Nella seguente tabella sono descritte importanti aggiunte alla documentazione di Amazon Cognito. Inoltre, effettuiamo aggiornamenti secondari frequenti alla documentazione in risposta al feedback inviato. Per inviare un feedback, individuare il collegamento Feedbackin fondo a qualsiasi pagina della documentazione di Amazon Cognito.

Modifica	Descrizione	Data
Aggiunto il supporto per oggetti complessi nel trigger Lambda pre token	Ora puoi aggiungere array e oggetti JSON alle attestazioni ID e Access Token.	30 maggio 2024
Informazioni aggiornate sulle autorizzazioni verificate e Amazon Cognito.	Amazon Verified Permissio ns ora ha un'integrazione più diretta con Amazon Cognito.	15 maggio 2024
Identità verificate in più regioni di Amazon SES.	In alcuni casi Regioni AWS senza Amazon SES, i pool di utenti di Amazon Cognito bilanciano il carico delle e-mail tra due regioni remote.	10 maggio 2024
Sono state aggiunte informazi oni sull'autorizzazione M2M e sulla gestione dei costi.	Scopri come utilizzare le concessioni di credenziali dei client per casi d'uso machine-t o-machine (M2M) con i pool di utenti di Amazon Cognito.	9 maggio 2024
Amazon Cognito è ora disponibile in Europa (Spagna) e Asia Pacifico (Hyderabad). Regioni AWS	Ora puoi creare risorse Amazon Cognito nelle regioni Europa (Spagna) e Asia Pacifico (Hyderabad).	15 aprile 2024
Amazon Cognito è ora disponibile nella regione Asia-	Ora puoi creare risorse Amazon Cognito nella regione Asia Pacifico (Melbourne).	4 aprile 2024

Pacifico (Melbourne). Regione AWS		
È stata aggiunta un'app Android di esempio in Flutter per i pool di utenti di Amazon Cognito.	Puoi creare un'app mobile iniziale per Amazon Cognito a partire da un'applicazione Flutter di esempio. GitHub	4 aprile 2024
Nuovi contenuti introduttivi	Contenuti estesi per iniziare, scenari comuni, best practice multi-tenant e accesso alle risorse dopo l'accesso.	1 aprile 2024
Amazon Cognito è ora disponibile in Europa (Zurigo). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione Europa (Zurigo).	14 marzo 2024
Amazon Cognito è ora disponibile in Medio Oriente (Emirati Arabi Uniti). Regione AWS	Ora puoi creare risorse Amazon Cognito nella regione del Medio Oriente (Emirati Arabi Uniti).	8 marzo 2024
Nuove funzionalità SAML e contenuti migliorati.	Ora puoi firmare le richieste SAML, crittografare le risposte SAML e configurare l'SSO SAML avviato dall'IdP.	1 febbraio 2024
Sono disponibili aumenti delle quote.	Ora puoi acquistare capacità aggiuntiva per le quote dei tassi di richiesta di Amazon Cognito.	25 gennaio 2024
I pool di identità di Amazon Cognito supportano le tariffe di richiesta in Service Quotas.	Ora puoi monitorare le quote requests-per-second (RPS) per i pool di identità di Amazon Cognito e richiedere un aumento nella console Service Quotas.	19 dicembre 2023

È stata aggiunta una nuova funzionalità per la personalizzazione del contenuto dei token di accesso.	È ora possibile aggiungere, modificare e rimuovere attestazioni e ambiti nei token di accesso del pool di utenti.	12 dicembre 2023
Contenuti migliorati sui client delle app e sugli ambiti OAuth.	Modifiche e correzioni di Clarity a Client dell'app pool di utenti e Autorizzazione Scopes, M2M e API con server di risorse. Istruzioni della console precedente rimosse.	14 novembre 2023
Contenuti migliorati sui dispositivi e sull'autenticazione dei dispositivi.	Nuovi contenuti sull'uso delle chiavi del dispositivo e sull'autenticazione SRP del dispositivo.	18 ottobre 2023
AWS Management Console Linee guida aggiornate.	È stato rimosso il riferimento alla console dei pool di utenti e gli argomenti sono stati ridistribuiti all'interno degli oggetti correlati e sono state aggiunte linee guida all'organizzazione basata su schede nella console Amazon Cognito.	30 agosto 2023
Accesso diretto minimizzato all'endpoint LOGIN.	È stata aggiunta una panoramica visiva del pool di utenti Endpoint Login ed enfatizzato l'avvio dell'autenticazione con Endpoint Authorize.	30 agosto 2023

Amazon Cognito è ora disponibile in Asia Pacifico (Osaka) e Israele (Tel Aviv). Regioni AWS	Ora puoi creare risorse Amazon Cognito nelle regioni di Asia Pacifico (Osaka) e Israele (Tel Aviv).	30 agosto 2023
Sono state introdotte informazioni sull'autorizzazione per Amazon Cognito con Amazon Verified Permissions.	Nella tua app, puoi richiamare l'API Autorizzazioni verificate per produrre decisioni di accesso da un'autorità centrale.	1° agosto 2023
È stata aggiunta una nuova funzionalità per la registrazione delle attività dettagliate degli utenti del pool di utenti su Amazon CloudWatch Logs.	Ora puoi registrare gli errori di recapito di e-mail e messaggi SMS nei gruppi di CloudWatch log.	1° agosto 2023
Informazioni aggiornate sulla politica AWS gestita per gli utenti guest del pool di identità.	L'ambito di applicazione delle autorizzazioni per gli utenti guest del pool di identità ora include sia una politica di sessione in linea che una politica di sessione gestita. AWS	16 maggio 2023
Miglioramento dei contenuti e nuove istruzioni della console per i pool di identità di Amazon Cognito.	Sono state aggiunte nuove procedure dettagliate sulla console per riflettere la nuova esperienza della console, dettagli sull'integrazione del codice migliorati per pool di identità.	16 maggio 2023
Aggiunte e miglioramenti alla home page del servizio e alla home page dei pool di utenti.	Pagine di panoramica aggiornate per Amazon Cognito e pool di utenti.	16 maggio 2023

Miglioramenti generali alla documentazione sui token del pool di utenti.	Sono stati aggiornati i token di esempio e sono state aggiunte nuove informazioni sulla verifica dei token.	16 febbraio 2023
Ora puoi registrare gli eventi relativi ai dati dei pool di identità di Amazon Cognito. AWS CloudTrail	CloudTrail supporta la selezione di pool di identità Amazon Cognito, operazioni API ad alto volume in percorsi che registrano gli eventi relativi ai dati.	15 febbraio 2023
Esempi e descrizioni aggiornati dei trigger Lambda.	Gli esempi di trigger Lambda vengono aggiornati alla JavaScript versione 3. Ora puoi correlare direttamente i trigger Lambda alle operazioni API.	31 gennaio 2023
I pool di identità di Amazon Cognito applicano una policy AWS gestita alle sessioni non autenticate.	Gli utenti del pool di identità che si autenticano utilizzando il flusso avanzato ora hanno una politica AWS gestita aggiuntiva applicata alla loro sessione.	31 gennaio 2023
Sono stati aggiunti esempi di codice.	Questa guida include ora un codice di esempio per l'app Amazon Cognito in una varietà di linguaggi di programmazione.	23 gennaio 2023
Sono state aggiunte informazioni sui modelli API e sull'autenticazione con i pool di utenti di Amazon Cognito.	I pool di utenti di Amazon Cognito dispongono di più interfacce API e formati per l'autorizzazione della richiesta.	15 dicembre 2022

Amazon Cognito è ora disponibile in Europa (Milano). Regione AWS	È ora possibile creare pool di utenti di Amazon Cognito nella regione Europa (Milano).	6 dicembre 2022
Sono state aggiunte informazioni sulla protezione dall'eliminazione del pool di utenti.	Quando crei un nuovo pool di utenti con AWS Management Console, ora è protetto dall'eliminazione per impostazione predefinita.	20 ottobre 2022
È stata aggiunta una guida per l'utente per l'interfaccia utente ospitata e informazioni sull'MFA TOTP nell'interfaccia utente ospitata.	Gli utenti ora possono registrare un dispositivo con autenticazione MFA con token di software TOTP nell'interfaccia utente ospitata di Amazon Cognito. È ora possibile visualizzare in anteprima l'interfaccia utente ospitata predefinita.	8 settembre 2022
Sono state aggiunte informazioni su AWS WAF Amazon Cognito.	Ora puoi associare un ACL AWS WAF Web a un pool di utenti Amazon Cognito.	3 agosto 2022
Sono stati aggiunti altri eventi di esempio AWS CloudTrail .	Amazon Cognito ora registra le richieste della federazione e dell'interfaccia utente ospitata nel tuo trail.	15 giugno 2022
Sono state aggiunte informazioni sulla verifica degli attributi in due passaggi.	Ora puoi scegliere se l'utente deve verificare un nuovo indirizzo e-mail o numero di telefono prima di poter eseguire l'accesso con esso.	9 giugno 2022

Documentazione federativa aggiornata. Nuova funzionalità di propagazione degli indirizzi IP.	Procedure dettagliate aggiornate per la configura zione dei social pool di utenti. IdPs Aggiunta di informazioni sui profili di utenti federati e sulla mappatura degli attributi . Sono state aggiunte nuove informazioni sulle impronte digitali dei dispositivi per una sicurezza avanzata.	31 maggio 2022
Accedi agli utenti federati senza interazione con l'interfa ccia utente ospitata	È stata aggiunta una nuova pagina su come aggiungere segnalibri alle applicazioni in modo che Amazon Cognito indirizzi silenziosamente gli utenti all'accesso federato.	29 maggio 2022
Messaggi SMS ed e-mail locali per pool di utenti Amazon Cognito	Ora puoi utilizzare Amazon Simple Notification Service per i messaggi SMS e Amazon Simple Email Service per i messaggi e-mail nello Regione AWS stesso pool di utenti.	14 marzo 2022
Aggiornamenti alla pagina delle quote	Sono state aggiunte e chiarite le quote relative alle risorse e ai tassi di richiesta.	10 gennaio 2022
Nuova esperienza di console con pool di utenti Amazon Cognito	Sono state aggiornate le istruzioni per creare e gestire i bacini d'utenza nella console Amazon Cognito aggiornata.	18 novembre 2021

RevokeToken API ed endpoint di revoca	È possibile utilizzare l' RevokeToken operazion e per revocare un token di aggiornamento per un utente.	10 giugno 2021
Best practice per più tenant	Sono state aggiunte le migliori pratiche per le applicazioni multi-tenant.	4 marzo 2021
Attributi per il controllo degli accessi	I pool di identità di Amazon Cognito forniscono attributi per il controllo degli accessi (AFAC) per consentire ai clienti di concedere agli utenti l'accesso alle risorse. AWS L'autorizzazione può essere eseguita in base agli attributi degli utenti dal provider di identità utilizzato per la federazione con Amazon Cognito.	15 gennaio 2021
Trigger Lambda del mittente SMS personalizzato e Trigger Lambda del mittente e-mail personalizzato	Il trigger Lambda del mittente SMS personalizzato e il trigger Lambda del mittente e-mail personalizzato consentono a un provider di terza parte di inviare notifiche e-mail e SMS agli utenti dall'interno del codice della funzione Lambda.	30 novembre 2020
Aggiornamenti del token Amazon Cognito	Sono state aggiunte informazi oni aggiornate sulla scadenza dei token di accesso, ID e aggiornamento.	29 ottobre 2020

Quotas del servizio Amazon Cognito	Le Service Quotas sono disponibili per le quote di categoria di Amazon Cognito. Puoi utilizzare la console Service Quotas per visualizzare l'utilizzo delle quote, richiedere un aumento delle quote e creare CloudWatch allarmi per monitorare l'utilizzo delle quote. Come parte di questa modifica, la sezione CloudWatch Metriche disponibili per i pool di utenti di Amazon Cognito è stata aggiornata per riflettere le nuove informazioni. Il nuovo nome della sezione è: Tracciamento delle quote e dell'utilizzo in CloudWatch e Service Quotas	29 ottobre 2020
Categorizzazione delle quote di Amazon Cognito	Per monitorare l'uso delle quote e richiederne un aumento sono disponibili le categorie di quote. Le quote sono raggruppate in categorie in base a casi d'uso comuni.	17 agosto 2020
Amazon Cognito è supportato da GovCloud negli Stati Uniti AWS	Amazon Cognito è ora supportato nella regione AWS GovCloud (Stati Uniti).	13 maggio 2020

Aggiornamenti dei documenti Amazon Cognito Pinpoint	È stato aggiunto un nuovo ruolo collegato ai servizi. Le istruzioni in "Utilizzo dell'analisi dei dati di Amazon Pinpoint con i bacini d'utenza di Amazon Cognito" sono state aggiornate.	13 maggio 2020
Nuovo capitolo dedicato alla sicurezza di Amazon Cognito	Il capitolo Sicurezza può aiutare la tua organizza zione a ottenere informazi oni approfondite sulla sicurezza dei servizi integrata e configurabile. AWS I nostri nuovi capitoli forniscono informazioni sulla sicurezza del cloud e nel cloud.	30 aprile 2020
Amazon Cognito Identity Pools ora supporta l'accesso con Apple	Accedi con Apple è disponibile in tutte le regioni in cui opera Amazon Cognito, eccetto nella regione cn-north-1.	7 aprile 2020
Nuovo controllo delle versioni dell'API di Facebook	Aggiunta la selezione della versione all'API di Facebook.	3 aprile 2020
Aggiornamento della differenz a tra maiuscole e minuscole	Aggiunta la raccomandazione sull'abilitazione della disattiva zione della distinzione tra maiuscole e minuscole del nomeutente prima di creare un bacino d'utenza.	11 febbraio 2020

Nuove informazioni su AWS Amplify	Sono state aggiunte informazioni sull'integrazione di Amazon Cognito con la tua app web o mobile AWS Amplify utilizzando SDK e librerie. Rimosse le informazioni sull'utilizzo degli SDK Amazon Cognito precedenti a AWS Amplify.	22 novembre 2019
Nuovo attributo per i trigger del pool di utenti	Amazon Cognito ora include un `clientMetadata` parametro nelle informazioni sugli eventi che trasmette alle AWS Lambda funzioni per la maggior parte dei trigger del pool di utenti. Puoi utilizzare questo parametro per migliorare il flusso di lavoro di autenticazione personalizzato con dati aggiuntivi.	4 ottobre 2019
Limite aggiornato	Il limite di limitazione per l'azione ListUsers API viene aggiornato.	25 giugno 2019
Nuovo limite	I limiti flessibili per i bacini d'utenza ora includono un limite per il numero di utenti.	17 giugno 2019

Impostazioni e-mail di Amazon SES per i pool di utenti Amazon Cognito	È possibile configurare un bacino d'utenza in modo che Amazon Cognito contatti via e-mail gli utenti tramite la configurazione di Amazon SES in uso. Questa impostazione consente ad Amazon Cognito di inviare e-mail con un volume di distribuzione più elevato di quanto altrimenti possibile.	8 Aprile 2019
Supporto per l'etichettatura	Aggiunte informazioni sull'assegnazione di tag alle risorse Amazon Cognito.	26 marzo 2019
Modifica il certificato per un dominio personalizzato	Se utilizzi un dominio personalizzato per l'interfaccia utente ospitata di Amazon Cognito, puoi modificare il certificato SSL per tale dominio in base alle esigenze.	19 dicembre 2018
Nuovo limite	È stato aggiunto un nuovo limite per il numero massimo di gruppi ai quali può appartenere ogni utente.	14 dicembre 2018
Limiti aggiornati	Sono stati aggiornati i limiti flessibili per i bacini d'utenza.	11 dicembre 2018
Aggiornamento della documentazione per la verifica degli indirizzi e-mail e dei numeri di telefono	Sono state aggiunte informazioni sulla configurazione del pool di utente per richiedere la verifica di e-mail o telefono quando un utente effettua la registrazione all'app.	20 novembre 2018

Aggiornamento della documentazione per testare le e-mail	Sono state aggiunte istruzioni per l'avvio delle e-mail da Amazon Cognito durante il test dell'app.	13 novembre 2018
Sicurezza avanzata Amazon Cognito	Sono state aggiunte nuove funzionalità di sicurezza per consentire agli sviluppatori di proteggere app e utenti da bot dannosi, garantire gli account utente se le credenziali sono state compromesse e regolare in modo automatico le sfide richieste per effettuare l'accesso in base al rischio calcolato del tentativo di accesso.	14 giugno 2018
Domini personalizzati per l'interfaccia utente ospitata da Amazon Cognito	Consente agli sviluppatori di utilizzare il proprio dominio completamente personalizzato per l'interfaccia utente ospitata in bacini d'utenza di Amazon Cognito.	4 giugno 2018
Pool di utenti di Amazon Cognito (OIDC Identity Provider)	Aggiunta della procedura di accesso al bacino d'utenza tramite un provider di identità OpenID Connect (OIDC), ad esempio Salesforce o Ping Identity.	17 maggio 2018
Trigger di migrazione Amazon Cognito Lambda	Aggiunte le pagine che descrivono la funzione trigger di migrazione Lambda	8 aprile 2018

Aggiornamento della Guida per gli sviluppatori di Amazon Cognito	Aggiunto il livello principale "Che cos'è Amazon Cognito" e "Nozioni di base su Amazon Cognito". Aggiunti inoltre scenari comuni e riorganiz zato il sommario dei bacini d'utenza. Aggiunta una nuova sezione "Nozioni di base sui bacini d'utenza di Amazon Cognito".	6 aprile 2018
Beta di sicurezza avanzata di Amazon Cognito	Sono state aggiunte nuove funzionalità di sicurezza per consentire agli sviluppatori di proteggere app e utenti da bot dannosi, garantire gli account utente se le credenzia li sono state compromesse altrove su Internet, oltre a regolare in modo automatico le sfide richieste per effettuar e l'accesso in base al rischio calcolato del tentativo di accesso.	28 novembre 2017
Integrazione con Amazon Pinpoint	È stata aggiunta la possibili tà di usare Amazon Pinpoint per fornire analisi dei dati per le app dei bacini d'utenza di Amazon Cognito e per migliorare i dati utente per le campagne di Amazon Pinpoint.	26 settembre 2017

Federazione e funzionalità integrate dell'interfaccia utente delle app dei pool di utenti di Amazon Cognito	È stata aggiunta la possibilità di consentire agli utenti di accedere al bacino d'utenza tramite Facebook, Google, Login with Amazon oppure attraverso un provider di identità SAML. È stata aggiunta una interfaccia utente dell'app integrata personalizzabile e il supporto di OAuth 2.0 con attestazioni personalizzate.	10 agosto 2017
Modifiche delle funzionalità relative alla conformità a HIPAA e PCI	È stata aggiunta la possibilità di consentire agli utenti di utilizzare un numero di telefono o un indirizzo e-mail come nome utente.	6 luglio 2017
Gruppi di utenti e funzionalità di controllo degli accessi basate sui ruoli	È stata aggiunta la possibilità amministrativa di creare e gestire i gruppi di utenti. Gli amministratori possono assegnare ruoli IAM agli utenti in base all'appartenenza al gruppo e alle regole create dall'amministratore.	15 dicembre 2016
Aggiornamento della documentazione	Esempi aggiornati che mostrano come utilizzare i AWS Lambda trigger con i pool di utenti.	27 novembre 2016
Aggiornamento della documentazione	Esempi di codice iOS aggiornati.	18 novembre 2016

Aggiornamento della documentazione	Sono state aggiunte informazioni sul flusso di conferma per gli account utente.	9 Novembre 2016
Funzionalità di creazione di account utente	È stata aggiunta la funzione di gestione per creare gli account utente attraverso la console di Amazon Cognito e l'API.	6 ottobre 2016
Funzionalità di importazione degli utenti	È stata aggiunta la funzionalità di importazione in blocco per i bacini d'utenza di Cognito. Utilizza questa funzionalità per migrare gli utenti dal tuo provider di identità esistente a un bacino d'utenza di Amazon Cognito.	1 settembre 2016
Disponibilità generale dei pool di utenti di Cognito	È stata aggiunta la funzionalità dei bacini d'utenza di Cognito. Utilizza questa funzionalità per creare e gestire una directory utente e per aggiungere la registrazione e l'accesso alla tua app per dispositivi mobili o applicazione Web tramite i bacini d'utenza.	28 luglio 2016
Supporto SAML	È stato aggiunto il supporto per l'autenticazione con provider di identità tramite Security Assertion Markup Language 2.0 (SAML 2.0).	23 giugno 2016
CloudTrail integrazione	Aggiunta integrazione con AWS CloudTrail.	18 febbraio 2016

Integrazione di eventi con Lambda	Consente di eseguire una AWS Lambda funzione in risposta a eventi importanti in Amazon Cognito.	9 aprile 2015
Flusso di dati verso Amazon Kinesis	Fornisce controllo e informazioni nei flussi di dati.	4 marzo 2015
Supporto OpenID Connect	Abilita il supporto per i provider di OpenID Connect.	23 novembre 2014
Sincronizzazione push	Abilita il supporto per la sincronizzazione push in modalità silenziosa.	6 Novembre 2014
È stato aggiunto il supporto per le identità autenticate dagli sviluppatori	Consente agli sviluppatori che possiedono propri sistemi di gestione di autenticazione e di identità ad essere trattati come provider di identità in Amazon Cognito.	29 settembre 2014
Disponibilità generale di Amazon Cognito		10 luglio 2014

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.