om/contact-us/?cmpid=docs_headercta_contactus)

n_service_name=Cognito&topic_url=https://docs.aws.amazon.com/cognito/latest/developerguide/authentication.html#amazon-cognito-user-pools-authentication-flow)

(https://docs.aws.amazon.com)

Get started (#) Service guides (#) Developer tools (#) AI resources (#)

Documentation (https://docs.aws.amazon.com/index.html) > Amazon Cognito (https://docs.aws.amazon.com/cognito/index.html) > Develo...

Authentication with Amazon Cognito user pools

PDF (/pdfs/cognito/latest/developerguide/cognito-dg.pdf#authentication) Focus mode

On this page

Related resources

Amazon Cognito user pools API Reference (https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/index.html)

AWS CLI commands for Amazon Cognito user pools (https://docs.aws.amazon.com/cli/latest/reference/cognito-idp/

SDKs & Tools (https://aws.amazon.com/tools/)

▼ Recommended tasks

How to

Configure authentication methods for managed login (https://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flows-selection-managedlogin.html)

Configure Amazon Cognito to authorize REST APIs (https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-integrate-with-cognito.html)

Learn about

Understand Cognito user pools API capabilities (https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/Welcome.html)

▶ Recently added to this guide

Did this page help you?

Yes No

users.html)

- User pool tokens (amazon-cognito-user-pools-using-tokens-with-identity-providers.html)

Provide feedback (https://docs.aws.amazon.com/forms/aws-doc-feedback?hidden_service_name=Cognito&topic_url=https://docs.aws.amazon.com/en_us/cognito/latest/developerguide/authentica

Amazon Cognito includes several methods to authenticate your users. Users can sign in with pas and WebAuthn passkeys. Amazon Cognito can send them a one-time password in an email or SN message. You can implement Lambda functions that orchestrate your own sequence of challeng responses. These are *authentication flows*. In authentication flows, users provide a secret and An Cognito verifies the secret, then issues JSON web tokens (JWTs) for applications to process with libraries. In this chapter, we'll talk about how to configure your user pools and app clients for va authentication flows in various application environments. You'll learn about options for the use hosted sign-in pages of managed login, and for building your own logic and front end in an AWS

All user pools, whether you have a domain or not, can authenticate users in the user pools API. I add a domain to your user pool, you can use the user pool endpoints (https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-userpools-server-contract-referenc The user pools API supports a variety of authorization models and request flows for API request

To verify the identity of users, Amazon Cognito supports authentication flows that incorporate challenge types in addition to passwords like email and SMS message one-time passwords and passkeys.

Topics

Implement authentication flows

Whether you're implementing managed login (./authentication-flows-selection-managedlogin.html) o custom-built application front end (./authentication-flows-selection-sdk.html) with an AWS SDK for authentication, you must configure your app client for the types of authentication that you wan implement. The following information describes setup for authentication flows in your app clien (./user-pool-settings-client-apps.html) and your application.

App client supported flows | **Implement flows in your application**

You can configure supported flows for your app clients in the Amazon Cognito console or wit API in an AWS SDK. After you configure your app client to support these flows, you can deplo them in your application.

The following procedure configures available authentication flows for an app client with the Amazon Cognito console.

To configure an app client for authentication flows (console)

1. Sign in to AWS and navigate to the Amazon Cognito user pools console (https://console.aws.amazon.com/cognito/v2/idp) . Choose a user pool or create a new one.
2. In your user pool configuration, select the **App clients** menu. Choose an app client or cre new one.
3. Under **App client information**, select **Edit**.
4. Under **App client flows**, choose the authentication flows that you want to support.

To configure an app client for authentication flows (API/SDK)

To configure available authentication flows for an app client with the Amazon Cognito API, se value of `ExplicitAuthFlows` in a CreateUserPoolClient (https://docs.aws.amazon.com/cognit user-identity-pools/latest/APIReference/API_CreateUserPoolClient.html#CognitoUserPools-CreateUserPoolClient-request-ExplicitAuthFlows) or UpdateUserPoolClient (https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/API_UpdateUserPoolClient.html#CognitoUserPools-UpdateUserPoolClient-request-ExplicitAuthFlows) request. The following is an example that provisions secure remote password (SRP) and choice-based authentication to a client.

```
"ExplicitAuthFlows": [
   "ALLOW_USER_AUTH",
   "ALLOW_USER_SRP_AUTH
]
```

When you configure app client supported flows, you can specify the following options and AI values.

App client flow support

Authentication flow	Compatibility	Console	API
Choice-based authentication (./authentication-flows-selection-sdk.html#authentication-flows-selection-choice)	Server-side, client-side	Select an authentication type at sign-in	ALLOW USER_ UTH
Sign-in with persistent passwords (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-pools-authentication-flow-methods-password)	Client-side	Sign in with username and password	ALLOW USER_ ASSWO D_AUT
Sign-in with persistent passwords and secure payload (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-pools-authentication-flow-methods-srp)	Server-side, client-side	Sign in with secure remote password (SRP)	ALLOW USER_ RP_AU H
Refresh tokens (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-	Server-side, client-side	Get new user tokens from existing	ALLOW REFRE

Authentication flow	Compatibility	Console	API
pools-authentication-flow-methods-refresh)		authenticated sessions	H_TOK N_AUT
Server-side authentication (./authentication-flows-public-server-side.html#amazon-cognito-user-pools-server-side-authentication-flow)	Server-side	Sign in with server-side administrative credentials	ALLOW ADMIN USER_ ASSWO D_AUT
Custom authentication (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-pools-authentication-flow-methods-custom)	Server-side and client-side custom-built applications. Not compatible with managed login.	Sign in with custom authentication flows from Lambda triggers	ALLOW CUSTO _AUTH

Things to know about authentication with user pools

Consider the following information in the design of your authentication model with Amazon Co user pools.

Authentication flows in managed login and the hosted UI

Managed login (./cognito-user-pools-managed-login.html) has more options for authentication th the classic hosted UI. For example, users can do passwordless and passkey authentication onl managed login.

Custom authentication flows only available in AWS SDK authentication

You can't do *custom authentication flows*, or custom authentication with Lambda triggers (./u pool-lambda-challenge.html) , with managed login or the classic hosted UI. Custom authenticati available in authentication with AWS SDKs (./authentication-flows-selection-sdk.html) .

Managed login for external identity provider (IdP) sign-in

You can't sign users in through third-party IdPs (./cognito-user-pools-identity-federation.html) in authentication with AWS SDKs (./authentication-flows-selection-sdk.html) . You must implement managed login or the classic hosted UI, redirect to IdPs, and then process the resulting authentication object with OIDC libraries in your application. For more information about ma login, see User pool managed login (./cognito-user-pools-managed-login.html) .

Passwordless authentication effect on other user features

Activation of passwordless sign-in with one-time passwords (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-pools-authentication-flow-methods-passwor or passkeys (./amazon-cognito-user-pools-authentication-flow-methods.html#amazon-cognito-user-p authentication-flow-methods-passkey) in your user pool and app client has an effect on user crea and migration. When passwordless sign-in is active:

1. Administrators can create users without passwords. The default invitation message templ changes to no longer include the `{###}` password placeholder. For more information, se Creating user accounts as administrator (./how-to-create-user-accounts.html) .
2. For SDK-based SignUp (https://docs.aws.amazon.com/cognito-user-identity-pools/latest/APIReference/API_SignUp.html) operations, users aren't required to supply a pas when they sign up. Managed login and the hosted UI require a password in the sign-up pa

even if passwordless authentication is permitted. For more information, see Signing up ar confirming user accounts (./signing-up-users-in-your-app.html) .

3. Users imported from a CSV file can sign in immediatelywith passwordless options, withou password reset, if their attributes include an email address or phone number for an availa passwordless sign-in option. For more information, see Importing users into user pools fr CSV file (./cognito-user-pools-using-import-tool.html) .
4. Passwordless authentication doesn't invoke the user migration Lambda trigger (./user-poo lambda-migrate-user.html) .
5. Users who sign in with a passwordless first factor can't add a multi-factor authentication (./user-pool-settings-mfa.html) factor to their session. Only password-based authentication support MFA.

Passkey relying party URLs can't be on the public suffix list

You can use domain names that you own, like `www.example.com`, as the relying party (RP) I your passkey configuration. This configuration is intended to support custom-built applicatio that run on domains that you own. The public suffix list (https://publicsuffix.org/) , or PSL, co protected high-level domains. Amazon Cognito returns an error when you attempt to set you URL to a domain on the PSL.

Topics

- Authentication session flow duration (#authentication-flow-session-duration)
- Lockout behavior for failed sign-in attempts (#authentication-flow-lockout-behavior)

Authentication session flow duration

Depending on the features of your user pool, you can end up responding to several challenges t `InitiateAuth` and `RespondToAuthChallenge` before your app retrieves tokens from Ama Cognito. Amazon Cognito includes a session string in the response to each request. To combine requests into an authentication flow, include the session string from the response to the previou request in each subsequent request. By default, your users have three minutes to complete each challenge before the session string expires. To adjust this period, change your app client **Authen flow session duration**. The following procedure describes how to change this setting in your ap configuration.

ⓘ **Note**

Authentication flow session duration settings apply to authentication with the Amazon Cognito user pools API. Managed login sets session duration to 3 minutes for multi-facto authentication and 8 minutes for password-reset codes.

Amazon Cognito console | **User pools API**

To configure app client authentication flow session duration (AWS Management Console)

1. From the **App integration** tab in your user pool, select the name of your app client from **App clients and analytics** container.
2. Choose **Edit** in the **App client information** container.
3. Change the value of **Authentication flow session duration** to the validity duration that y want, in minutes, for SMS and email MFA codes. This also changes the amount of time th any user has to complete any authentication challenge in your app client.

4. Choose **Save changes**.

For more information about app clients, see Application-specific settings with app clients (./user-settings-client-apps.html) .

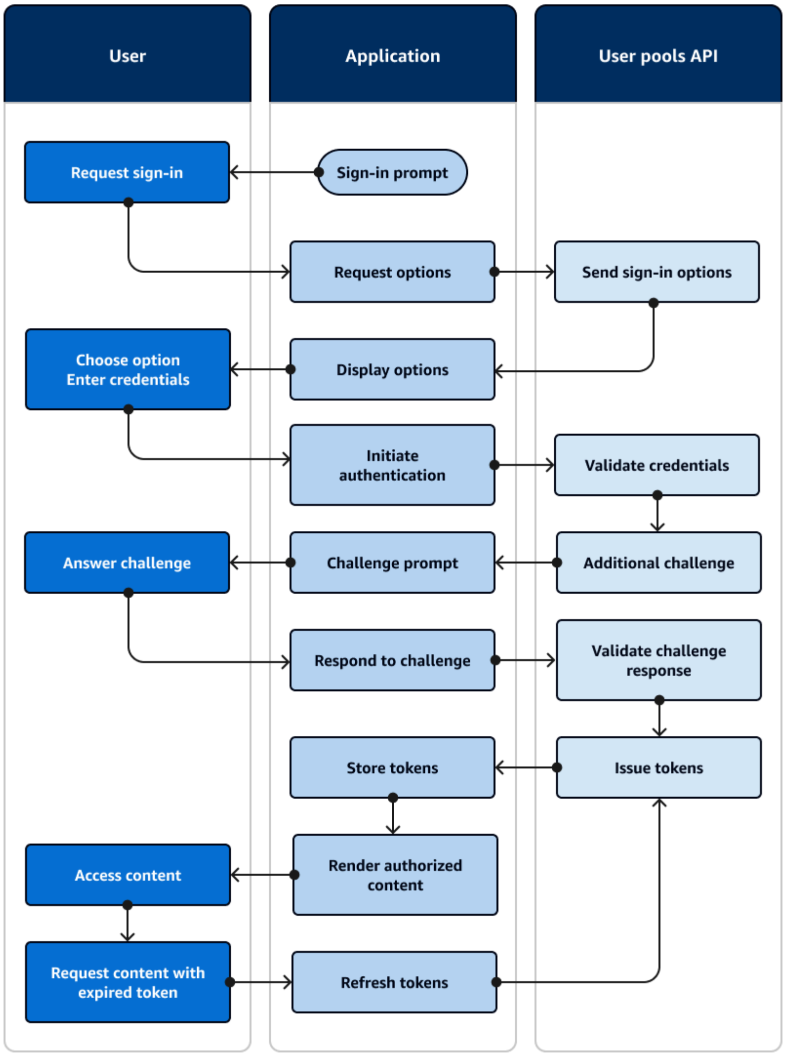
Lockout behavior for failed sign-in attempts

After five failed sign-in attempts with a user's password, regardless of whether those are reques unauthenticated or IAM-authorized API operations, Amazon Cognito locks out your user for one The lockout duration then doubles after each additional one failed attempt, up to a maximum o approximately 15 minutes.

Attempts made during a lockout period generate a `Password attempts exceeded` exceptio don't affect the duration of subsequent lockout periods. For a cumulative number of failed sign-attempts *n*, not including `Password attempts exceeded` exceptions, Amazon Cognito locks your user for *2^(n-5)* seconds. To reset the lockout to its *n=0* initial state, your user must either s successfully after a lockout period expires, or not initiate any sign-in attempts for 15 consecutiv minutes at any time after a lockout. This behavior is subject to change. This behavior doesn't ap custom challenges unless they also perform password-based authentication.

An example authentication session

The following diagram and step-by-step guide illustrate a typical scenario where a user signs in application. The example application presents a user with several sign-in options. They select or entering their credentials, provide an additional authentication factor, and sign in.



Picture an application with a sign-in page where users can sign in with a username and passwor request a one-time code in an email message, or choose a fingerprint option.

1. **Sign-in prompt**: Your application shows a home screen with a *Log in* button.
2. **Request sign-in**: The user selects *Log in*. From a cookie or a cache, your application retrieves username, or prompts them to enter it.
3. **Request options**: Your application requests the user's sign-in options with an `InitiateAu` request with the `USER_AUTH` flow, requesting the available sign-in methods for the user.
4. **Send sign-in options**: Amazon Cognito responds with `PASSWORD`, `EMAIL_OTP`, and `WEB_/` The response includes a session identifier for you to replay back in the next response.
5. **Display options**: Your application shows UI elements for the user to enter their username a password, get a one-time code, or scan their fingerprint.
6. **Choose option/Enter credentials**: The user enters their username and password.
7. **Initiate authentication**: Your application provides the user's sign-in information with a `RespondToAuthChallenge` API request that confirms username-password sign-in and pr the username and the password.
8. **Validate credentials**: Amazon Cognito confirms the user's credentials.

9. **Additional challenge**: The user has multi-factor authentication configured with an authenti app. Amazon Cognito returns a `SOFTWARE_TOKEN_MFA` challenge.
10. **Challenge prompt**: Your application displays a form requesting a time-based one-time pass (TOTP) from the user's authenticator app.
11. **Answer challenge**: The user submits the TOTP.
12. **Respond to challenge**: In another `RespondToAuthChallenge` request, your application p the user's TOTP.
13. **Validate challenge response**: Amazon Cognito confirms the user's code and determines tha user pool is configured to issue no additional challenges to the current user.
14. **Issue tokens**: Amazon Cognito returns ID, access, and refresh JSON web tokens (JWTs). The initial authentication is complete.
15. **Store tokens**: Your application caches the user's tokens so that it can reference user data, au access to resources, and update tokens when they expire.
16. **Render authorized content**: Your application makes a determination of the user's access to resources based on their identity and roles, and delivers application content.
17. **Access content**: The user is signed in and begins using the application.
18. **Request content with expired token**: Later, the user requests a resource that requires authorization. The user's cached token has expired.
19. **Refresh tokens**: Your application makes an `InitiateAuth` request with the user's saved re token.
20. **Issue tokens**: Amazon Cognito returns new ID and access JWTs. The user's session is securely refreshed without additional prompts for credentials.

You can use AWS Lambda triggers (./cognito-user-pools-working-with-lambda-triggers.html) to custo the way users authenticate. These triggers issue and verify their own challenges as part of the authentication flow.

You can also use the admin authentication flow for secure backend servers. You can use the user migration authentication flow (./cognito-user-pools-using-import-tool.html) to make user migration possible without the requirement that your users to reset their passwords.

View related pages ✦ *Abstracts generated by AI*

Cognito › developerguide
Manage authentication methods in AWS SDKs...
Manage authentication methods including one-time passwords, WebAuthn passkeys, persistent passwords, refresh tokens, and custom authentication in AWS SDKs.
October 27, 2025

Cognito › developerguide
Authentication flows (https://docs.aws.amazon.c...
Amazon Cognito user pools offer various authentication flows, including passwordless sign-in with one-time passwords or WebAuthn passkeys.
October 27, 2025

Prescriptive-guidance › patte
Send custom attribut Amazon Cognito and
October 26, 2025

Discover highly rated pages ✦ *Abstracts generated by AI*

Cognito › developerguide
What is Amazon Cognito? (https://docs.aws.amazon.c...
Amazon Cognito authenticates users, authorizes AWS resource access, issues temporary AWS credentials, integrates with identity providers, manages user pools and identity pools, configures role-based access control.
October 26, 2025

Cognito › developerguide
Understanding user pool JSON web tokens (JWTs)...
Authenticating users with tokens, storing tokens securely, customizing access and ID tokens, understanding ID token claims, understanding access token claims, refreshing tokens for new access, revoking tokens to end sessions.
October 26, 2025

Cognito › developerguide
Common Amazon Cog scenarios...
Amazon Cognito enables authentication, access to end resources, AWS servi via API Gateway, Lambda identity pools, third-par IdPs, and AppSync resou
October 26, 2025

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.