Temporary security credentials in IAM

Archived: 2026-04-05 23:49:37 UTC

You can use the AWS Security Token Service (AWS STS) to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to long-term access key credentials, with the following differences:

- Temporary security credentials are *short-term*, as the name implies. They can be configured to last for anywhere from a few minutes to several hours. After the credentials expire, AWS no longer recognizes them or allows any kind of access from API requests made with them.
- Temporary security credentials are not stored with the user but are generated dynamically and provided to the user when requested. When (or even before) the temporary security credentials expire, the user can request new credentials, as long as the user requesting them still has permissions to do so.

As a result, temporary credentials have the following advantages over long-term credentials:

- You do not have to distribute or embed long-term AWS security credentials with an application.
- You can provide access to your AWS resources to users without having to define an AWS identity for them. Temporary credentials are the basis for roles and identity federation.
- The temporary security credentials have a limited lifetime, so you do not have to update them or explicitly revoke them when they're no longer needed. After temporary security credentials expire, they cannot be reused. You can specify how long the credentials are valid, up to a maximum limit.

AWS STS and AWS regions

Temporary security credentials are generated by AWS STS. By default, AWS STS is a global service with a single endpoint at `https://sts.amazonaws.com`. However, you can also choose to make AWS STS API calls to endpoints in any other supported Region. This can reduce latency (server lag) by sending the requests to servers in a Region that is geographically closer to you. No matter which Region your credentials come from, they work globally. For more information, see Manage AWS STS in an AWS Region.

Common scenarios for temporary credentials

Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles.

Identity federation

You can manage your user identities in an external system outside of AWS and grant users who sign in from those systems access to perform AWS tasks and access your AWS resources. IAM supports two types of identity

federation. In both cases, the identities are stored outside of AWS. The distinction is where the external system resides—in your data center or an external third party on the web. To compare features of temporary security credentials for identity federation, see Compare AWS STS credentials.

For more information about external identity providers, see Identity providers and federation into AWS.

- **OpenID Connect (OIDC) federation** – You can let users sign in using a well-known third-party identity provider such as Login with Amazon, Facebook, Google, or any OIDC 2.0 compatible provider for your mobile or web application, you don't need to create custom sign-in code or manage your own user identities. Using OIDC federation helps you keep your AWS account secure, because you don't have to distribute long-term security credentials, such as IAM user access keys, with your application. For more information, see OIDC federation.

  AWS STS OIDC federation supports Login with Amazon, Facebook, Google, and any OpenID Connect (OIDC)-compatible identity provider.

  Note

  For mobile applications, we recommend that you use Amazon Cognito. You can use this service with AWS SDKs for mobile development to create unique identities for users and authenticate them for secure access to your AWS resources. Amazon Cognito supports the same identity providers as AWS STS, and also supports unauthenticated (guest) access and lets you migrate user data when a user signs in. Amazon Cognito also provides API operations for synchronizing user data so that it is preserved as users move between devices. For more information, see Authentication with Amplify in the *Amplify Documentation*.

- **SAML federation** – You can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with different sign-in credentials. This is known as the *single sign-on* approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your own solution for federating user identities. For more information, see SAML 2.0 federation.

  - **Custom federation broker** – You can use your organization's authentication system to grant access to AWS resources. For an example scenario, see Enable custom identity broker access to the AWS console.

  - **Federation using SAML 2.0** – You can use your organization's authentication system and SAML to grant access to AWS resources. For more information and an example scenario, see SAML 2.0 federation.

Roles for cross-account access

Many organizations maintain more than one AWS account. Using roles and cross-account access, you can define user identities in one account, and use those identities to access AWS resources in other accounts that belong to your organization. This is known as the *delegation* approach to temporary access. For more information about

creating cross-account roles, see Create a role to give permissions to an IAM user. To learn whether principals in accounts outside of your zone of trust (trusted organization or account) have access to assume your roles, see What is IAM Access Analyzer?.

Roles for Amazon EC2

If you run applications on Amazon EC2 instances and those applications need access to AWS resources, you can provide temporary security credentials to your instances when you launch them. These temporary security credentials are available to all applications that run on the instance, so you don't need to store any long-term credentials on the instance. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances.

To learn more about IAM Amazon EC2 role credentials, see IAM roles for Amazon EC2 in the *Amazon Elastic Compute Cloud User Guide*.

Other AWS services

You can use temporary security credentials to access most AWS services. For a list of the services that accept temporary security credentials, see AWS services that work with IAM.

Sample applications that use temporary credentials

You can use AWS Security Token Service (AWS STS) to create and provide trusted users with temporary security credentials that can control access to your AWS resources. For more information about AWS STS, see Temporary security credentials in IAM. To see how you can use AWS STS to manage temporary security credentials, you can download the following sample applications that implement complete example scenarios:

- Enabling Federation to AWS Using Windows Active Directory, ADFS, and SAML 2.0. Demonstrates how to delgate access using enterprise federation to AWS using Windows Active Directory (AD), Active Directory Federation Services (ADFS) 2.0, and SAML (Security Assertion Markup Language) 2.0.
- Enable custom identity broker access to the AWS console. Demonstrates how to create a custom federation proxy that enables single sign-on (SSO) so that existing Active Directory users can sign in to the AWS Management Console.
- How to Use Shibboleth for Single Sign-On to the AWS Management Console.. Shows how to use Shibboleth and SAML to provide users with single sign-on (SSO) access to the AWS Management Console.

Samples for OIDC federation

The following sample applications illustrate how to use OIDCfederation with providers like Login with Amazon, Amazon Cognito, Facebook, or Google. You can trade authentication from these providers for temporary AWS security credentials to access AWS services.

- Amazon Cognito Tutorials – We recommend that you use Amazon Cognito with the AWS SDKs for mobile development. Amazon Cognito is the simplest way to manage identity for mobile apps, and it provides additional features like synchronization and cross-device identity. For more information about Amazon Cognito, see Authentication with Amplify in the *Amplify Documentation*.

The following scenarios and applications can guide you in using temporary security credentials:

- How to integrate AWS STS SourceIdentity with your identity provider. This post shows you how to set up the AWS STS `SourceIdentity` attribute when using Okta, Ping, or OneLogin as your IdP.
- OIDC federation. This section discusses how to configure IAM roles when you use OIDC federation and the `AssumeRoleWithWebIdentity` API.
- Secure API access with MFA. This topic explains how to use roles to require multi-factor authentication (MFA) to protect sensitive API actions in your account.

For more information on policies and permissions in AWS see the following topics:

- Access management for AWS resources
- Policy evaluation logic.
- Managing Access Permissions to Your Amazon S3 Resources in *Amazon Simple Storage Service User Guide*.
- To learn whether principals in accounts outside of your zone of trust (trusted organization or account) have access to assume your roles, see What is IAM Access Analyzer?.
