



User Guide

AWS IAM Identity Center



AWS IAM Identity Center: User Guide

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is IAM Identity Center?	1
Why use IAM Identity Center?	1
IAM Identity Center rename	3
Legacy namespaces remain the same	3
Enabling IAM Identity Center	4
Prerequisites and considerations	6
Considerations for choosing an AWS Region	6
Using IAM Identity Center for user access to applications only	8
Quota for IAM roles created by IAM Identity Center	8
IAM Identity Center and AWS Organizations	9
Confirm your identity sources in IAM Identity Center	10
Getting started tutorials for IAM Identity Center	13
Active Directory	14
CyberArk	16
Prerequisites	17
SCIM considerations	17
Step 1: Enable provisioning in IAM Identity Center	18
Step 2: Configure provisioning in CyberArk	19
(Optional) Step 3: Configure user attributes in CyberArk for access control (ABAC) in IAM Identity Center	20
(Optional) Passing attributes for access control	21
Google Workspace	21
JumpCloud	31
Prerequisites	32
SCIM considerations	32
Step 1: Enable provisioning in IAM Identity Center	32
Step 2: Configure provisioning in JumpCloud	33
(Optional) Step 3: Configure user attributes in JumpCloud for access control in IAM Identity Center	34
(Optional) Passing attributes for access control	35
Microsoft Entra ID	36
Okta	52
OneLogin	61
Prerequisites	62

Step 1: Enable provisioning in IAM Identity Center	62
Step 2: Configure provisioning in OneLogin	63
(Optional) Step 3: Configure user attributes in OneLogin for access control in IAM Identity Center	64
(Optional) Passing attributes for access control	65
Troubleshooting	65
Ping Identity	66
PingFederate	67
PingOne	73
Identity Center directory	79
Common tasks	85
Set up access to applications	86
Create a permission set for job functions	89
Create a permission set that applies least-privilege permissions	90
Assign user access	92
Assign group access	93
View user and group assignments	96
Sign in to the AWS access portal	96
Manage instances	99
Organization instances of IAM Identity Center	101
When to use an organization instance	101
Account instances of IAM Identity Center	101
Availability constraints for member accounts	102
When to use account instances	103
Account instance considerations	103
Supported AWS managed applications	104
Enable account instance creation	104
Control account instance creation	105
Create an account instance	106
Delete your IAM Identity Center instance	107
Authentication in IAM Identity Center	109
Authentication sessions	109
.....	110
Connect workforce users	111
IAM Identity Center use cases	111
Enable single sign-on access to your AWS applications	112

Enable single sign-on access to your Amazon EC2 Windows instances	113
Users, groups, and provisioning	114
User name and email address uniqueness	114
Groups	114
User and group provisioning	115
Manage your identity source	115
Considerations for changing your identity source	116
Change your identity source	121
Manage sign-in and attribute use for all identity source types	122
Manage identities in IAM Identity Center	128
Connect to a Microsoft AD directory	139
Manage an external identity provider	162
Using the AWS access portal	176
Activating the AWS access portal	177
Signing in to the AWS access portal	177
Resetting your user password	179
AWS CLI and AWS SDK access	180
Creating shortcut links	185
Registering your device for MFA	188
Customizing the AWS access portal URL	190
Multi-factor authentication	190
Available MFA types	191
Configure MFA	194
Manage MFA	201
Manage access to AWS accounts	204
AWS account types	204
Assigning AWS account access	207
End-user experience	207
Enforcing and limiting access	208
Delegating and enforcing access	208
Limiting access to the identity store from member accounts	208
Delegated administration	209
Best practices	210
Prerequisites	210
Register a member account	211
Deregister a member account	212

View which member account has been registered as the delegated administrator	213
Temporary elevated access	213
Validated AWS Security Partners for temporary elevated access	214
Temporary elevated access capabilities assessed for AWS partner validation	215
Single sign-on access to AWS accounts	216
Assign user access to AWS accounts	216
Remove user and group access to an AWS account	219
Revoke an active permission set session	219
Delegate who can assign single sign-on access	221
Permission sets	222
Predefined permissions	223
Custom permissions	224
Create, manage, and delete permission sets	227
Configure permission set properties	235
Referencing permission sets in resource policies, Amazon EKS Cluster config maps, and AWS	
KMS key policies	241
Recommendations to avoid access disruptions	243
Custom trust policy example	244
Attribute-based access control	245
Benefits	246
Checklist: Configuring ABAC in AWS using IAM Identity Center	247
Attributes for access control	249
Repair the IAM identity provider	256
Service-linked roles	256
Manage access to applications	257
AWS managed applications	257
Controlling access	262
Coordinating administrative tasks	262
Configuring IAM Identity Center to share identity information	263
Considerations for sharing identity information in AWS accounts	263
Enabling identity-aware console sessions	264
Constraining the use of AWS managed applications	267
Viewing and updating application details	267
Disabling an AWS managed application	268
Customer managed applications	269
SAML 2.0 and OAuth 2.0 applications	270

SAML 2.0 application setup	274
Trusted identity propagation	277
Overview	278
Use cases	279
Set up trusted identity propagation	291
Trusted token issuer	306
Manage certificates	318
Considerations before rotating a certificate	318
Rotate an IAM Identity Center certificate	319
Certificate expiration status indicators	321
Application properties	321
Application start URL	322
Relay state	322
Session duration	323
Assign user access to applications	324
Remove user access	325
Map attributes	325
Resiliency design and Regional behavior	327
Designed for availability	328
Set up emergency access to the AWS Management Console	328
Overview	328
Summary of emergency access configuration	329
How to design your critical operations roles	330
How to plan your access model	331
How to design emergency role, account, and group mapping	332
How to create your emergency access configuration	332
Emergency preparation tasks	334
Emergency failover process	334
Return to normal operations	335
One-time setup of a direct IAM federation application in Okta	335
Security	338
Identity and access management for IAM Identity Center	339
Authentication	339
Access control	339
Overview of managing access	340
Identity-based policies (IAM policies)	343

AWS managed policies	351
Using service-linked roles	369
IAM Identity Center console and API authorization	376
API actions after November 2023	377
API actions after October 2020	378
AWS STS condition keys for IAM Identity Center	380
UserId	381
IdentityStoreArn	381
ApplicationArn	381
CredentialId	382
InstanceArn	382
Logging and monitoring	382
Logging IAM Identity Center API calls with AWS CloudTrail	383
Amazon EventBridge	408
Logging AD sync and configurable AD sync errors	408
Compliance validation	411
Supported compliance standards	412
Resilience	414
Infrastructure security	414
Tagging resources	416
Tag restrictions	417
Managing tags with the console	417
AWS CLI examples	418
Assigning tags	418
Viewing tags	419
Removing tags	419
Applying tags when you create a permission set	419
API actions	420
API actions for IAM Identity Center instance tags	420
Integrating AWS CLI with IAM Identity Center	421
How to integrate AWS CLI with IAM Identity Center	421
Region availability	422
IAM Identity Center Region data	422
Choosing your Region	422
Cross-Region calls	422
Managing IAM Identity Center in an opt-in Region	424

Disabling an AWS Region where IAM Identity Center is enabled	424
Quotas	426
Application quotas	426
AWS account quotas	427
Active Directory quotas	428
IAM Identity Center identity store quotas	428
IAM Identity Center throttle limits	428
Additional quotas	429
Troubleshooting	430
Issues creating an account instance of IAM Identity Center	430
You receive an error when you attempt to view the list of cloud applications that are preconfigured to work with IAM Identity Center	430
Issues regarding contents of SAML assertions created by IAM Identity Center	431
Specific users fail to synchronize into IAM Identity Center from an external SCIM provider	432
Users can't sign in when their user name is in UPN format	433
I get a 'Cannot perform the operation on the protected role' error when modifying an IAM role	434
Directory users cannot reset their password	434
My user is referenced in a permission set but can't access the assigned accounts or applications	435
I can't get my application from the application catalog configured correctly	435
Error 'An unexpected error has occurred' when a user tries to sign in using an external identity provider	436
Error 'Attributes for access control failed to enable'	437
I get a 'Browser not supported' message when I attempt to register a device for MFA	437
Active Directory "Domain Users" group does not properly sync into IAM Identity Center	437
Invalid MFA credentials error	437
I get a 'An unexpected error has occurred' message when I attempt to register or sign in using an authenticator app	438
I get an 'It's not you, it's us' error when attempting to sign in to IAM Identity Center	438
My users are not receiving emails from IAM Identity Center	439
Error: You can't delete/modify/remove/assign access to permission sets provisioned in the management account	439
Error: Session token not found or invalid	439
Document history	440
AWS Glossary	447

What is IAM Identity Center?

AWS IAM Identity Center is the AWS solution for connecting your workforce users to AWS managed applications such as Amazon Q Developer and Amazon QuickSight, and other AWS resources. You can connect your existing identity provider and synchronize users and groups from your directory, or create and manage your users directly in IAM Identity Center. You can then use IAM Identity Center for either or both of the following:

- User access to applications
- User access to AWS accounts

Already using IAM for access to AWS accounts?

You don't need to make any changes to your current AWS account workflows to use IAM Identity Center for access to AWS managed applications. If you're using [federation with IAM](#) or IAM users for AWS account access, your users can continue to access AWS accounts in the same way they always have, and you can continue to use your existing workflows to manage that access.

Why use IAM Identity Center?

IAM Identity Center streamlines and simplifies workforce user access to applications or AWS accounts, or both, through the following key capabilities.

Integration with AWS managed applications

[AWS managed applications](#) such as Amazon Q Developer and Amazon Redshift integrate with IAM Identity Center. IAM Identity Center provides AWS managed applications with a common view of users and groups.

Trusted identity propagation across applications

With trusted identity propagation, AWS managed applications such as Amazon QuickSight can securely share a user's identity with other AWS managed applications such as Amazon Redshift and authorize access to AWS resources based on the user's identity. You can more easily audit user activity because CloudTrail events are logged based on the user and the actions the user initiated. This makes it easier to understand who accessed what. For information about supported use cases, including end-to-end configuration guidance, see [Trusted identity propagation use cases](#).

One place to assign permissions to multiple AWS accounts

With multi-account permissions, IAM Identity Center provides a single place for you to assign permissions to groups of users in multiple AWS accounts. You can create permissions based on common job functions or define custom permissions that meet your security needs. You can then assign those permissions to workforce users to control their access to specific AWS accounts.

This optional feature is available only for [organization instances](#) of IAM Identity Center.

One point of federation to simplify user access to AWS

By providing one point of federation, IAM Identity Center reduces the administrative effort required to use multiple AWS managed applications and AWS accounts. With IAM Identity Center, you only federate once, and you have only one certificate to manage when using a [SAML 2.0](#) identity provider. IAM Identity Center provides AWS managed applications with a common view of users and groups for trusted identity propagation use cases, or when users share access to AWS resources with other people.

For information about how to configure commonly used identity providers to work with IAM Identity Center, see [Getting started tutorials for IAM Identity Center](#). If you don't have an existing identity provider, you can [create and manage users directly in IAM Identity Center](#).

Two modes of deployment

IAM Identity Center supports two types of instances: *organization instances* and *account instances*. An organization instance is the best practice. It's the only instance that enables you to manage access to AWS accounts and it's recommended for all production use of applications. An organization instance is deployed in the AWS Organizations management account and gives you a single point from which to manage user access across AWS.

Account instances are bound to the AWS account in which they are enabled. Use account instances of IAM Identity Center only to support isolated deployments of select AWS managed applications. For more information, see [Manage organization and account instances of IAM Identity Center](#).

User-friendly web portal access for your users

The AWS access portal is a user-friendly web portal that provides your users with seamless access to all their assigned applications, AWS accounts, or both.

IAM Identity Center rename

On July 26, 2022, AWS Single Sign-On was renamed to AWS IAM Identity Center.

Legacy namespaces remain the same

The `sso` and `identitystore` API namespaces along with the following related namespaces **remain unchanged** for backward compatibility purposes.

- CLI commands
 - [aws configure sso](#)
 - [identitystore](#)
 - [sso](#)
 - [sso-admin](#)
 - [sso-oidc](#)
- [Managed policies](#) containing `AWSSSO` and `AWSIdentitySync` prefixes
- [Service endpoints](#) containing `sso` and `identitystore`
- [AWS CloudFormation](#) resources containing `AWS::SSO` prefixes
- [Service-linked role](#) containing `AWSServiceRoleForSSO`
- Console URLs containing `sso` and `singlesignon`
- Documentation URLs containing `singlesignon`

Enabling AWS IAM Identity Center

Complete the following steps to sign in to the AWS Management Console and enable an [organization instance](#) of IAM Identity Center.

1. Do either of the following to sign in to the AWS Management Console.
 - **New to AWS (root user)** – Sign in as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
 - **Already using AWS (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. Open the [IAM Identity Center console](#).
3. Under **Enable IAM Identity Center**, choose **Enable with AWS Organizations**.
4. *Optional* Add tags that you want to associate with this organization instance.
5. *Optional* Configure delegated administration.

Note

If you are using a multi-account environment, we recommend that you configure delegated administration. With delegated administration, you can limit the number of people who require access to the management account in AWS Organizations. For more information, see [Delegated administration](#).

Important

The ability to create [account instances of IAM Identity Center](#) is enabled by default. Account instances of IAM Identity Center include a subset of features available to an organization instance. You can control whether [users can access this feature](#) by using a Service Control Policy.

Do you need to update firewalls and gateways?

If you filter access to specific AWS domains or URL endpoints by using a web content filtering solution such as next-generation firewalls (NGFW) or Secure Web Gateways (SWG), you must add

the following domains or URL endpoints to your web-content filtering solution allowlists. Doing so enables you to access your AWS access portal.

- *[Directory ID or alias].awsapps.com*
- *.aws.dev
- *.awsstatic.com
- *.console.aws.a2z.com
- oidc.*[Region]*.amazonaws.com
- *.sso.amazonaws.com
- *.sso.*[Region]*.amazonaws.com
- *.sso-portal.*[Region]*.amazonaws.com
- *[Region]*.signin.aws
- *[Region]*.signin.aws.amazon.com
- signin.aws.amazon.com
- *.cloudfront.net
- opfcaptcha-prod.s3.amazonaws.com

Considerations for allowlisting domains and URL endpoints

Understand the impact of allowlisting domains beyond AWS access portal.

- To access AWS accounts, the AWS Management Console, and the IAM Identity Center console from your AWS access portal, you must allowlist additional domains. Refer to [Troubleshooting](#) in the *AWS Management Console Getting Started Guide* for a list of AWS Management Console domains.
- To access AWS managed applications from your AWS access portal, you must allowlist their respective domains. Refer to the respective service documentation for guidance.
- These allowlists cover AWS services. If you use external software, such as external IdPs (for example, Okta and Microsoft Entra ID), you'll need to include their domains in your allowlists.

You are now ready to configure IAM Identity Center. When you enable IAM Identity Center it's automatically configured with an Identity Center directory as your default identity source, which is the fastest way to get started using IAM Identity Center. For instructions, see [Configure user access with the default IAM Identity Center directory](#).

If you want to learn more about how IAM Identity Center works with Organizations, identity sources, and IAM roles, see the following topics.

Topics

- [Prerequisites and considerations](#)
- [Confirm your identity sources in IAM Identity Center](#)

Prerequisites and considerations

The following topics provide information about prerequisites and other considerations for setting up IAM Identity Center.

Topics

- [Considerations for choosing an AWS Region](#)
- [Using IAM Identity Center for user access to applications only](#)
- [Quota for IAM roles created by IAM Identity Center](#)
- [IAM Identity Center and AWS Organizations](#)

Considerations for choosing an AWS Region

You can enable an IAM Identity Center instance in a single, supported AWS Region of your choice. Choosing a Region requires an assessment of your priorities based on your use cases and company policies. Access to AWS accounts and cloud applications from your IAM Identity Center doesn't depend on this choice; however, access to AWS managed applications and the ability to use AWS Managed Microsoft AD as the identity source can depend on this choice. Refer to [AWS IAM Identity Center endpoints and quotas](#) in the AWS General Reference for a list of Regions that IAM Identity Center supports.

Key considerations for choosing an AWS Region.

- **Geographical location** – When you select a Region that is geographically closest to the majority of your end users, they'll have lower latency of access to the AWS access portal and AWS managed applications, such as Amazon SageMaker Studio.
- **Availability of AWS managed applications** – AWS managed applications, such as Amazon SageMaker, can operate only in the AWS Regions they support. Enable IAM Identity Center in a Region supported by the AWS managed application(s) you want to use with it. Many AWS

managed applications can also operate only in the same Region where you enabled IAM Identity Center.

- **Digital sovereignty** – Digital sovereignty regulations or company policies may mandate the use of a particular AWS Region. Consult with your company’s legal department.
- **Identity source** – If you’re using AWS Managed Microsoft AD or AD Connector as the identity source, its home Region must match the AWS Region in which you enabled IAM Identity Center.
- **Regions disabled by default** – AWS originally enabled all new AWS Regions for use in AWS accounts by default, which automatically enabled your users to create resources in any Region. Now when AWS adds a new Region, its use is disabled by default in all accounts. If you deploy IAM Identity Center in a Region disabled by default, then you must enable this Region in all the accounts for which you want to manage access to IAM Identity Center. This is required even if you don’t plan to create any resources in that Region in those accounts.

You can enable a Region for the current accounts in your organization and you must repeat this action for new accounts you might add later. For instructions, see [Enable or disable a Region in your organization](#) in the AWS Organizations user guide. To avoid repeating these additional steps, you can choose to deploy your IAM Identity Center in a Region enabled by default. For reference, the following Regions are enabled by default:

- US East (Ohio)
- US East (N. Virginia)
- US West (Oregon)
- US West (N. California)
- Europe (Paris)
- South America (São Paulo)
- Asia Pacific (Mumbai)
- Europe (Stockholm)
- Asia Pacific (Seoul)
- Asia Pacific (Tokyo)
- Europe (Ireland)
- Europe (Frankfurt)
- Europe (London)
- Asia Pacific (Singapore)

- Canada (Central)
- Asia Pacific (Osaka)
- **Cross-Region calls** – In some Regions, IAM Identity Center may call Amazon Simple Email Service in a different Region to send email. In these cross-Region calls, IAM Identity Center sends certain user attributes to the other Region. For more information about Regions, see [AWS IAM Identity Center Region availability](#).

Switching AWS Regions

You can switch your IAM Identity Center Region only by deleting the current instance and creating a new instance in another Region. If you already enabled an AWS managed application with your existing instance, you should delete it first before deleting your IAM Identity Center. You must recreate users, groups, permission sets, applications, and assignments in the new instance. You can use the IAM Identity Center account and application assignment APIs to get a snapshot of your configuration and then use that snapshot to rebuild your configuration in a new Region. You may also need to recreate some IAM Identity Center configuration through the Management Console of your new instance. For instructions on deleting IAM Identity Center, see [Delete your IAM Identity Center instance](#).

Using IAM Identity Center for user access to applications only

You can use IAM Identity Center for user access to applications such as Amazon Q Developer, AWS accounts, or both. You can connect your existing identity provider and synchronize users and groups from your directory, or [create and manage users directly in IAM Identity Center](#). For information about how to connect your existing identity provider to IAM Identity Center, see the [Getting started tutorials for IAM Identity Center](#).

Already using IAM for access to AWS accounts?

You don't need to make any changes to your current AWS account workflows to use IAM Identity Center for access to AWS managed applications. If you're using [federation with IAM](#) or IAM users for AWS account access, your users can continue to access AWS accounts in the same way they always have, and you can continue to use your existing workflows to manage that access.

Quota for IAM roles created by IAM Identity Center

IAM Identity Center creates IAM roles to give users permissions to resources. When you assign a permission set, IAM Identity Center creates corresponding IAM Identity Center-controlled IAM

roles in each account, and attaches the policies specified in the permission set to those roles. IAM Identity Center manages the role, and allows the authorized users you've defined to assume the role, by using the AWS access portal or AWS CLI. As you modify the permission set, IAM Identity Center ensures that the corresponding IAM policies and roles are updated accordingly.

If you've already configured IAM roles in your AWS account, we recommend that you check whether your account is approaching the quota for IAM roles. The default quota for IAM roles per account is 1000 roles. For more information, see [IAM object quotas](#).

If you're nearing the quota, consider requesting a quota increase. Otherwise, you might experience problems with IAM Identity Center when you provision permission sets to accounts that have exceeded the IAM role quota. For information about how to request a quota increase, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

Note

If you are reviewing IAM roles in an account that's already using IAM Identity Center, you might notice role names beginning with "AWSReservedSSO_". These are the roles which the IAM Identity Center service has created in the account, and they came from assigning a permission set to the account.

IAM Identity Center and AWS Organizations

AWS Organizations is recommended, but not required, for use with IAM Identity Center. If you haven't set up an organization, you don't have to. When you enable IAM Identity Center, you will choose whether to enable the service with AWS Organizations. When you set up an organization, the AWS account that sets up the organization becomes the management account of the organization. The root user of the AWS account is now the owner of the organizational management account. Any additional AWS accounts you invite to your organization are member accounts. The management account creates the organizations resources, organizational units, and policies that manage the member accounts. Permissions are delegated to member accounts by the management account.

Note

We recommend that you enable IAM Identity Center with AWS Organizations, which creates an organization instance of IAM Identity Center. An organization instance is our

recommended best practice because it supports all features of IAM Identity Center and provides central management capabilities. For more information, see [Manage organization and account instances of IAM Identity Center](#).

If you've already set up AWS Organizations and are going to add IAM Identity Center to your organization, make sure that all AWS Organizations features are enabled. When you create an organization, enabling all features is the default. For more information, see [Enabling all features in your organization](#) in the *AWS Organizations User Guide*.

To enable IAM Identity Center, you must sign in to the AWS Management Console by signing in to your AWS Organizations management account as a user that has administrative credentials or as the root user (not recommended unless no other administrative users exist). You can't enable IAM Identity Center while signed in with administrative credentials from an AWS Organizations member account. For more information, see [Creating and managing an AWS Organization](#) in the *AWS Organizations User Guide*.

For more information on managing your AWS Organizations, see the following:

- [Deleting an organization](#)
- [Considerations before removing an account from an organization](#)

Confirm your identity sources in IAM Identity Center

Your identity source in IAM Identity Center defines where your users and groups are managed. After you enable IAM Identity Center, confirm that you're using the identity source of your choice. If you already have an assigned identity source, you can continue to use it.

Confirm your identity source

1. Open the [IAM Identity Center console](#).
2. On the **Dashboard** page, below the **Recommended setup steps** section, choose **Confirm your identity source**. You can also access this page by choosing **Settings** and choosing the **Identity source** tab.
3. There is no action if you want to keep your assigned identity source. If you prefer to change it, choose **Actions**, and then choose **Change identity source**.

You can choose one of the following as your identity source:

Identity Center directory

When you enable IAM Identity Center for the first time, it's automatically configured with an Identity Center directory as your default identity source. If you aren't already using another external identity provider, you can get started creating your users and groups, and assign their level of access to your AWS accounts and applications. For a tutorial on using this identity source, see [Configure user access with the default IAM Identity Center directory](#).

Active Directory

If you're already managing users and groups in either your AWS Managed Microsoft AD directory using AWS Directory Service or your self-managed directory in Active Directory (AD), we recommend that you connect that directory when you enable IAM Identity Center. Don't create any users and groups in the default Identity Center directory. IAM Identity Center uses the connection provided by the AWS Directory Service to synchronize user, group, and membership information from your source directory in Active Directory to the IAM Identity Center identity store. For more information, see [Connect to a Microsoft AD directory](#).


Note

IAM Identity Center doesn't support SAMBA4-based Simple AD as an identity source.

External identity provider

For external identity providers (IdPs) such as Okta or Microsoft Entra ID, you can use IAM Identity Center to authenticate identities from the IdPs through the Security Assertion Markup Language (SAML) 2.0 standard. The SAML protocol doesn't provide a way to query the IdP to learn about users and groups. You make IAM Identity Center aware of those users and groups by provisioning them into IAM Identity Center. You can perform automatic provisioning (synchronization) of user and group information from your IdP into IAM Identity Center using the System for Cross-domain Identity Management (SCIM) v2.0 protocol if your IdP supports SCIM. Otherwise, you can manually provision your users and groups by manually entering the user names, email address, and groups into IAM Identity Center.

For detailed instructions on setting up your identity source, see [Getting started tutorials for IAM Identity Center](#).

 **Note**

If you plan to use an external identity provider, note that the external IdP, not IAM Identity Center, manages multi-factor authentication (MFA) settings. MFA in IAM Identity Center isn't supported for use by external identity providers. For more information, see [Prompt users for MFA](#).

The identity source that you choose determines where IAM Identity Center searches for users and groups that need single sign-on access. After you confirm or change your identity source, you'll create or specify a user and assign them administrative permissions to your AWS account.

 **Important**

If you're already managing users and groups in Active Directory or an external IdP, we recommend that you consider connecting this identity source when you enable IAM Identity Center and choose your identity source. This should be done before you create any users and groups in the default Identity Center directory and make any assignments.

If you're already managing users and groups in one identity source in IAM Identity Center, changing to a different identity source might remove all user and group assignments that you configured in IAM Identity Center. If this occurs, all users, including the administrative user in IAM Identity Center, will lose single sign-on access to their AWS accounts and applications. For more information, see [Considerations for changing your identity source](#).

After you configure your identity source, you can look up users or groups to grant them single sign-on access to AWS accounts, cloud applications, or both.

Getting started tutorials for IAM Identity Center

You can connect your existing identity source in your AWS Organizations management account to [an organization instance of IAM Identity Center](#). If you don't have an existing identity provider, you can create and manage users directly in the default IAM Identity Center directory. You can have one identity source per organization.

The tutorials in this section describe how to set up an organization instance of IAM Identity Center with a commonly used identity source, create an administrative user, and if you're using IAM Identity Center to manage access to AWS accounts, create and configure permission sets. If you're using IAM Identity Center for application access only, you do not need to use permission sets.

These tutorials do not describe how to set up account instances of IAM Identity Center. You can use account instances to assign users and groups to applications, but you cannot use this instance type to manage user access to AWS accounts. For more information, see [Account instances of IAM Identity Center](#).

Note

Before starting any of these tutorials, enable IAM Identity Center. For more information, see [Enabling AWS IAM Identity Center](#).

Topics

- [Using Active Directory as an identity source](#)
- [Setting up SCIM provisioning between CyberArk and IAM Identity Center](#)
- [Configure SAML and SCIM with Google Workspace and IAM Identity Center](#)
- [Using IAM Identity Center to connect with your JumpCloud Directory Platform](#)
- [Configure SAML and SCIM with Microsoft Entra ID and IAM Identity Center](#)
- [Configure SAML and SCIM with Okta and IAM Identity Center](#)
- [Setting up SCIM provisioning between OneLogin and IAM Identity Center](#)
- [Using Ping Identity products with IAM Identity Center](#)
- [Configure user access with the default IAM Identity Center directory](#)

Using Active Directory as an identity source

If you are managing users in either your AWS Managed Microsoft AD directory using AWS Directory Service or your self-managed directory in Active Directory (AD), you can change your IAM Identity Center identity source to work with those users. We recommend that you consider connecting this identity source when you enable IAM Identity Center and choose your identity source. Doing this before you create any users and groups in the default Identity Center directory will help you avoid the additional configuration that's required if you change your identity source later.

To use Active Directory as your identity source, your configuration must meet the following prerequisites:

- If you're using AWS Managed Microsoft AD, you must enable IAM Identity Center in the same AWS Region where your AWS Managed Microsoft AD directory is set up. IAM Identity Center stores the assignment data in the same Region as the directory. To administer IAM Identity Center, you might need to switch to the Region where IAM Identity Center is configured. Also, note that the AWS access portal uses the same access URL as your directory.
- Use an Active Directory residing in the management account:

You must have an existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory Service, and it must reside within your AWS Organizations management account. You can connect only one AD Connector directory or one directory in AWS Managed Microsoft AD at a time. If you need to support multiple domains or forests, use AWS Managed Microsoft AD. For more information, see:

- [Connect a directory in AWS Managed Microsoft AD to IAM Identity Center](#)
- [Connect a self-managed directory in Active Directory to IAM Identity Center](#)
- Use an Active Directory residing in the delegated administrator account:

If you plan to enable an IAM Identity Center delegated administrator and use Active Directory as your IAM Identity Center identity source, you can use an existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory residing in the delegated admin account.

If you decide to change the IAM Identity Center identity source from any other source to Active Directory, or change it from Active Directory to any other source, the directory must reside in (be owned by) the IAM Identity Center delegated administrator member account if one exists; otherwise, it must be in the management account.

This tutorial guides you through the basic set up for using Active Directory as an IAM Identity Center identity source.

Step 1: Connect Active Directory and specify a user

If you're already using Active Directory, the following topics will help you prepare to connect your directory to IAM Identity Center.

Note

As a security best practice, we strongly recommend that you enable multi-factor authentication. If you plan to connect an AWS Managed Microsoft AD directory or a self-managed directory in Active Directory and you're not using RADIUS MFA with AWS Directory Service, enable MFA in IAM Identity Center.

AWS Managed Microsoft AD

1. Review the guidance in [Connect to a Microsoft AD directory](#).
2. Follow the steps in [Connect a directory in AWS Managed Microsoft AD to IAM Identity Center](#).
3. Configure Active Directory to synchronize the user to whom you want to grant administrative permissions into IAM Identity Center. For more information, see [Synchronize an administrative user into IAM Identity Center](#).

Self-managed directory in Active Directory

1. Review the guidance in [Connect to a Microsoft AD directory](#).
2. Follow the steps in [Connect a self-managed directory in Active Directory to IAM Identity Center](#).
3. Configure Active Directory to synchronize the user to whom you want to grant administrative permissions into IAM Identity Center. For more information, see [Synchronize an administrative user into IAM Identity Center](#).

Step 2: Synchronize an administrative user into IAM Identity Center

After you connect your directory to IAM Identity Center, you can specify a user to whom you want to grant administrative permissions, and then synchronize that user from your directory into IAM Identity Center.

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. On the **Manage Sync** page, choose the **Users** tab, and then choose **Add users and groups**.
5. On the **Users** tab, under **User**, enter the exact username and choose **Add**.
6. Under **Added Users and Groups**, do the following:
 - a. Confirm that the user to whom you want to grant administrative permissions is specified.
 - b. Select the check box to the left of the username.
 - c. Choose **Submit**.
7. In the **Manage sync** page, the user that you specified appears in the **Users in sync scope** list.
8. In the navigation pane, choose **Users**.
9. On the **Users** page, it might take some time for the user that you specified to appear in the list. Choose the refresh icon to update the list of users.

At this point, your user doesn't have access to the management account. You will set up administrative access to this account by creating an administrative permission set and assigning the user to that permission set. For more information, see [Create a permission set for job functions](#).

Setting up SCIM provisioning between CyberArk and IAM Identity Center

IAM Identity Center supports automatic provisioning (synchronization) of user information from CyberArk Directory Platform into IAM Identity Center. This provisioning uses the System for Cross-domain Identity Management (SCIM) v2.0 protocol. You configure this connection in CyberArk using your IAM Identity Center SCIM endpoint and access token. When you configure SCIM synchronization, you create a mapping of your user attributes in CyberArk to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and CyberArk.

This guide is based on CyberArk as of August 2021. Steps for newer versions may vary. This guide contains a few notes regarding configuration of user authentication through SAML.

Note

Before you begin deploying SCIM, we recommend that you first review the [Considerations for using automatic provisioning](#). Then continue reviewing additional considerations in the next section.

Topics

- [Prerequisites](#)
- [SCIM considerations](#)
- [Step 1: Enable provisioning in IAM Identity Center](#)
- [Step 2: Configure provisioning in CyberArk](#)
- [\(Optional\) Step 3: Configure user attributes in CyberArk for access control \(ABAC\) in IAM Identity Center](#)
- [\(Optional\) Passing attributes for access control](#)

Prerequisites

You will need the following before you can get started:

- CyberArk subscription or free trial. To sign up for a free trial visit [CyberArk](#).
- An IAM Identity Center enabled account ([free](#)). For more information, see [Enable IAM Identity Center](#).
- A SAML connection from your CyberArk account to IAM Identity Center, as described in [CyberArk documentation for IAM Identity Center](#).
- Associate the IAM Identity Center connector with the roles, users and organizations you want to allow access to AWS accounts.

SCIM considerations

The following are considerations when using CyberArk federation for IAM Identity Center:

- Only roles mapped in the application Provisioning section will be synchronized to IAM Identity Center.

- The provisioning script is supported only in its default state, once changed the SCIM provisioning might fail.
 - Only one phone number attribute can be synchronized and the default is “work phone”.
- If the role mapping in CyberArk IAM Identity Center application is changed, the below behavior is expected:
 - If the role names are changed - no changes to the group names in IAM Identity Center.
 - If the group names are changed - new groups will be created in IAM Identity Center, old groups will remain but will have no members.
- User synchronization and de-provisioning behavior can be set up from the CyberArk IAM Identity Center application, make sure you set up the right behavior for your organization. These are the options you have:
 - Overwrite (or not) users in Identity Center directory with the same principal name.
 - De-provision users from IAM Identity Center when the user is removed from the CyberArk role.
 - De-provision user behavior - disable or delete.

Step 1: Enable provisioning in IAM Identity Center

In this first step, you use the IAM Identity Center console to enable automatic provisioning.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

⚠ Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

Now that you have set up provisioning in the IAM Identity Center console, you need to complete the remaining tasks using the CyberArk IAM Identity Center application. These steps are described in the following procedure.

Step 2: Configure provisioning in CyberArk

Use the following procedure in the CyberArk IAM Identity Center application to enable provisioning with IAM Identity Center. This procedure assumes that you have already added the CyberArk IAM Identity Center application to your CyberArk admin console under **Web Apps**. If you have not yet done so, refer to the [Prerequisites](#), and then complete this procedure to configure SCIM provisioning.

To configure provisioning in CyberArk

1. Open the CyberArk IAM Identity Center application that you added as part of configuring SAML for CyberArk (**Apps > Web App**). See [Prerequisites](#).
2. Choose the **IAM Identity Center** application and go to the **Provisioning** section.
3. Check the box for **Enable provisioning for this application** and choose **Live Mode**.
4. In the previous procedure, you copied the **SCIM endpoint** value from IAM Identity Center. Paste that value into the **SCIM Service URL** field, in the CyberArk IAM Identity Center application set the **Authorization Type** to be **Authorization Header**.
5. Set the **Header Type** to **Bearer Token**.
6. From the previous procedure you copied the **Access token** value in IAM Identity Center. Paste that value into the **Bearer Token** field in the CyberArk IAM Identity Center application.
7. Click **Verify** to test and apply the configuration.

8. Under the **Sync Options**, choose the right behavior for which you want the outbound provisioning from CyberArk to work. You can choose to overwrite (or not) existing IAM Identity Center users with similar principal name, and the de-provisioning behavior.
9. Under **Role Mapping** set up the mapping from CyberArk roles, under the **Name** field to the IAM Identity Center group, under the **Destination Group**.
10. Click **Save** at the bottom once you are done.
11. To verify that users have been successfully synchronized to IAM Identity Center, return to the IAM Identity Center console and choose **Users**. Synchronized users from CyberArk will appear on the **Users** page. These users can now be assigned to accounts and can connect within IAM Identity Center.

(Optional) Step 3: Configure user attributes in CyberArk for access control (ABAC) in IAM Identity Center

This is an optional procedure for CyberArk should you choose to configure attributes for IAM Identity Center to manage access to your AWS resources. The attributes that you define in CyberArk are passed in a SAML assertion to IAM Identity Center. You then create a permission set in IAM Identity Center to manage access based on the attributes you passed from CyberArk.

Before you begin this procedure, you must first enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

To configure user attributes in CyberArk for access control in IAM Identity Center

1. Open the CyberArk IAM Identity Center application that you installed as part of configuring SAML for CyberArk (**Apps > Web Apps**).
2. Go to the **SAML Response** option.
3. Under **Attributes**, add the relevant attributes to the table following the below logic:
 - a. **Attribute Name** is the original attribute name from CyberArk.
 - b. **Attribute Value** is the attribute name sent in the SAML assertion to IAM Identity Center.
4. Choose **Save**.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Configure SAML and SCIM with Google Workspace and IAM Identity Center

If your organization is using Google Workspace you can integrate your users from Google Workspace into IAM Identity Center to give them access to AWS resources. You can achieve this integration by changing your IAM Identity Center identity source from the default IAM Identity Center identity source to Google Workspace.

User information from Google Workspace is synchronized into IAM Identity Center using the [System for Cross-domain Identity Management \(SCIM\) 2.0 protocol](#). You configure this connection in Google Workspace using your SCIM endpoint for IAM Identity Center and an IAM Identity Center bearer token. When you configure SCIM synchronization, you create a mapping of your user attributes in Google Workspace to the named attributes in IAM Identity Center. This mapping matches the expected user attributes between IAM Identity Center and Google Workspace. To do this, you need to set up Google Workspace as an IAM identity provider and an IAM Identity Center identity provider.

Objective

The steps in this tutorial help guide you through establishing the SAML connection between Google Workspace and AWS. Later, you will synchronize users from Google Workspace using SCIM. To verify everything is configured correctly, after completing the configuration steps you will sign in as a Google Workspace user and verify access to AWS resources. Note that this tutorial is based on a small Google Workspace directory test environment. Directory structures such as groups and organization units aren't included in this tutorial. After completing this tutorial, your users will be able to access the AWS access portal with your Google Workspace credentials.

Note

To sign up for a free trial of Google Workspace visit [Google Workspace](#) on Google's website. If you haven't enabled IAM Identity Center yet, see [Enabling AWS IAM Identity Center](#).

Considerations

- Before you configure SCIM provisioning between Google Workspace and IAM Identity Center, we recommend that you first review [Considerations for using automatic provisioning](#).
- SCIM automatic synchronization from Google Workspace is currently limited to user provisioning. Automatic group provisioning is not supported at this time. Groups can be manually created with AWS CLI Identity Store [create-group](#) command or AWS Identity and Access Management (IAM) API [CreateGroup](#). Alternatively, you can use [ssosync](#) to synchronize Google Workspace users and groups into IAM Identity Center.
- Every Google Workspace user must have a **First name**, **Last name**, **Username** and **Display name** value specified.
- Each Google Workspace user has only a single value per data attribute, such as email address or phone number. Any users that have multiple values will fail to synchronize. If there are users that have multiple values in their attributes, remove the duplicate attributes before attempting to provision the user in IAM Identity Center. For example, only one phone number attribute can be synchronized, since the default phone number attribute is "work phone", use the "work phone" attribute to store the user's phone number, even if the phone number for the user is a home phone or a mobile phone.
- Attributes are still synchronized if the user is disabled in IAM Identity Center, but still active in Google Workspace.
- If there is an existing user in Identity Center directory with the same username and email, the user will be overwritten and synchronized using SCIM from Google Workspace.

- There are additional considerations when changing your identity source. For more information, see [the section called “Changing from IAM Identity Center to an external IdP”](#).

Step 1: Google Workspace: Configure the SAML application

1. Sign in to your **Google Admin console** using an account with super administrator privileges.
2. In the left navigation panel of your **Google Admin console**, choose **Apps** and then choose **Web and Mobile Apps**.
3. In the **Add app** dropdown list, select **Search for apps**.
4. In the search box enter **Amazon Web Services**, then select **Amazon Web Services (SAML)** app from the list.
5. On the **Google Identity Provider details - Amazon Web Services** page, you can do either of the following:
 - a. Download IdP metadata.
 - b. Copy the SSO URL, Entity ID URL, and Certificate information.

You will need either the XML file or URL information in Step 2.

6. Before moving to the next step in the Google Admin console, leave this page open and move to the IAM Identity Center console.

Step 2: IAM Identity Center and Google Workspace: Change the IAM Identity Center identity source and setup Google Workspace as an SAML identity provider

1. Sign in to the [IAM Identity Center console](#) using a role with administrative permissions.
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, choose **Actions**, and then choose **Change identity source**.
 - If you haven't enabled IAM Identity Center, see [Enabling IAM Identity Center](#) for more information. After enabling and accessing IAM Identity Center for the first time, you will arrive at the **Dashboard** where you can select **Choose your identity source**.
4. On the **Choose identity source** page, select **External identity provider**, and then choose **Next**.
5. The **Configure external identity provider** page opens. To complete this page and the Google Workspace page in Step 1, you will need to complete the following:

- Under **Identity Provider metadata** section in the **IAM Identity Center** console, you will need to do either of the following:
 - i. Upload the **Google SAML metadata** as the **IdP SAML metadata** in the IAM Identity Center console.
 - ii. Copy and paste the **Google SSO URL** into the **IdP Sign-in URL** field, **Google Issuer URL** into the **IdP issuer URL** field, and upload the **Google Certificate** as the **IdP certificate**.
6. After providing the Google metadata in the **Identity Provider metadata** section of the **IAM Identity Center** console, copy the **IAM Identity Assertion Consumer Service (ACS) URL** and **IAM Identity Center issuer URL**. You will need to provide these URLs in the Google Admin console in the next step.
 7. Leave the page open with the IAM Identity Center console and return to the Google Admin console. You should be on the **Amazon Web Services - Service Provider details** page. Select **Continue**.
 8. On the **Service provider details** page, enter the **ACS URL** and **Entity ID** values. You copied these values in the previous step and they can be found in the IAM Identity Center console.
 - Paste the **IAM Identity Center Assertion Consumer Service (ACS) URL** into the **ACS URL** field
 - Paste the **IAM Identity Center issuer URL** into the **Entity ID** field.
 9. On the **Service provider details** page, complete the fields under **Name ID** as follows:
 - For **Name ID format**, select **EMAIL**
 - For **Name ID**, select **Basic Information > Primary email**
 10. Choose **Continue**.
 11. On the **Attribute Mapping** page, under **Attributes**, choose **ADD MAPPING**, and then configure these fields under **Google Directory attribute**:
 - For the `https://aws.amazon.com/SAML/Attributes/RoleSessionName` **app attribute**, select the field **Basic Information, Primary Email** from the **Google Directory attributes**.
 - For the `https://aws.amazon.com/SAML/Attributes/Role` **app attribute**, select any **Google Directory attributes**. A Google Directory attribute could be **Department**.
 12. Choose **Finish**

13. Return to the **IAM Identity Center** console and choose **Next**. On the **Review and Confirm** page, review the information and then enter **ACCEPT** into the space provided. Choose **Change identity source**.

You are now ready to enable the Amazon Web Services app in Google Workspace so that your users can be provisioned into IAM Identity Center.

Step 3: Google Workspace: Enable the apps

1. Return to the **Google Admin Console** and your AWS IAM Identity Center application which can be found under **Apps** and **Web and Mobile Apps**.
2. In the **User access** panel next to **User access**, choose the down arrow to expand **User access** to display the **Service status** panel.
3. In **Service status** panel, choose **ON for everyone**, and then choose **SAVE**.

Note

To help maintain the principle of least privilege, we recommend that after you complete this tutorial you change the **Service status** to **OFF for everyone**. Only users that need access to AWS should have the service enabled. You can use Google Workspace groups or organizational units to give user access to a particular subset of your users.

Step 4: IAM Identity Center: Set up IAM Identity Center automatic provisioning

1. Return to the IAM Identity Center console.
2. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
3. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. In Step 5 of this tutorial, you will enter these values to configure automatic provisioning in Google Workspace.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

⚠ Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward.


4. Choose **Close.**

Now that you've set up provisioning in the IAM Identity Center console, in the next step you will configure auto provisioning in Google Workspace.

Step 5: Google Workspace: Configure auto provisioning


1. Return to the Google Admin console and your AWS IAM Identity Center application which can be found under **Apps and Web and Mobile apps**. In the **Auto provisioning** section, choose **Configure auto provisioning**.
2. In the previous procedure, you copied the **Access token** value in IAM Identity Center console. Paste that value into the **Access token** field and choose **Continue**. Also, in the previous procedure, you copied the **SCIM endpoint** value in IAM Identity Center console. Paste that value into the **Endpoint URL** field and choose **Continue**.
3. Verify that all mandatory IAM Identity Center attributes (those marked with *) are mapped to Google Cloud Directory attributes. If not, choose the down arrow and map to the appropriate attribute. Choose **Continue**.
4. In **Provisioning scope** section, you can choose a group with your Google Workspace directory to provide access to the Amazon Web Services app. Skip this step and select **Continue**.
5. In **Deprovisioning** section, you can choose how to respond to different events that remove access from a user. For each situation you can specify the amount of time before deprovisioning begins to:
 - within 24 hours
 - after one day
 - after seven days
 - after 30 days

Each situation has a time setting for when to suspend an account's access and when to delete the account.

 **Tip**

Always set more time before deleting a user's account than for suspending a user's account.

6. Choose **Finish**. You are returned to the Amazon Web Services app page.
7. In the **Auto-provisioning** section, turn on the toggle switch to change it from **Inactive** to **Active**.

 **Note**

The activation slider is disabled if IAM Identity Center isn't turned on for users. Choose **User access** and turn the app on to enable the slider.

8. In the confirmation dialog box, choose **Turn on**.
9. To verify that users are successfully synchronized to IAM Identity Center, return to the IAM Identity Center console and choose **Users**. The **Users** page lists the users from your Google Workspace directory that were created by SCIM. If users aren't listed yet, it might be that provisioning is still in process. Provisioning can take up to 24 hours, although in most cases it completes within minutes. Make sure to refresh the browser window every few minutes.

Select a user and view their details. The information should match the information in the Google Workspace directory.

 **Congratulations!**

You have successfully set up a SAML connection between Google Workspace and AWS and have verified that automatic provisioning is working. You can now assign these users to accounts and applications in **IAM Identity Center**. For this tutorial, in the next step let's designate one of the users as the IAM Identity Center administrator by granting them administrative permissions to the management account.


Step 6: IAM Identity Center: Grant Google Workspace users access to accounts

1. Return to the **IAM Identity Center** console. In the IAM Identity Center navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
2. On the **AWS accounts** page the **Organizational structure** displays your organizational root with your accounts underneath it in the hierarchy. Select the checkbox for your management account, then select **Assign users or groups**.
3. The **Assign users and groups** workflow displays. It consists of three steps:
 - a. For **Step 1: Select users and groups** choose the user that will be performing the administrator job function. Then choose **Next**.
 - b. For **Step 2: Select permission sets** choose **Create permission set** to open a new tab that steps you through the three sub-steps involved in creating a permission set.
 - i. For **Step 1: Select permission set type** complete the following:
 - In **Permission set type**, choose **Predefined permission set**.
 - In **Policy for predefined permission set**, choose **AdministratorAccess**.Choose **Next**.
 - ii. For **Step 2: Specify permission set details**, keep the default settings, and choose **Next**.

The default settings create a permission set named *AdministratorAccess* with session duration set to one hour.
 - iii. For **Step 3: Review and create**, verify that the **Permission set type** uses the AWS managed policy **AdministratorAccess**. Choose **Create**. On the **Permission sets** page a notification appears informing you that the permission set was created. You can close this tab in your web browser now.
 - iv. On the **Assign users and groups** browser tab, you are still on **Step 2: Select permission sets** from which you started the create permission set workflow.
 - v. In the **Permissions sets** area, choose the **Refresh** button. The *AdministratorAccess* permission set you created appears in the list. Select the checkbox for that permission set and then choose **Next**.
 - c. For **Step 3: Review and submit** review the selected user and permission set, then choose **Submit**.

The page updates with a message that your AWS account is being configured. Wait until the process completes.

You are returned to the AWS accounts page. A notification message informs you that your AWS account has been reprovisioned and the updated permission set applied. When the user sign in they will have the option of choosing the *AdministratorAccess* role.

 **Note**

SCIM automatic synchronization from Google Workspace only supports provisioning users. Automatic group provisioning is not supported at this time. You can't create groups for your Google Workspace users using the AWS Management Console. After provisioning users, you can create groups using AWS CLI Identity Store [create-group](#) command or IAM API [CreateGroup](#).

Step 7: Google Workspace: Confirm Google Workspace users access to AWS resources

1. Sign in to Google using a test user account. To learn how to add users to Google Workspace, see [Google Workspace documentation](#).
2. Select the Google apps launcher (waffle) icon.
3. Scroll to the bottom of the apps list where your custom Google Workspace apps are located. The **Amazon Web Services** app is displayed.
4. Select the **Amazon Web Services** app. You are signed into the AWS access portal and can see the AWS account icon. Expand that icon to see the list of AWS accounts that the user can access. In this tutorial you only worked with a single account, so expanding the icon only shows one account.
5. Select the account to display the permission sets available to the user. In this tutorial you created the **AdministratorAccess** permission set.
6. Next to the permission set are links for the type of access available for that permission set. When you created the permission set, you specified both management console and programmatic access be enabled, so those two options are present. Select **Management console** to open the AWS Management Console.
7. The user is signed in to the console.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Next steps

Now that you've configured Google Workspace as an identity provider and provisioned users in IAM Identity Center, you can:

- Use the AWS CLI Identity Store [create-group](#) command or IAM API [CreateGroup](#) to create groups for your users.

Groups are useful when assigning access to AWS accounts and applications. Rather than assign each user individually, you give permissions to a group. Later, as you add or remove users from a group, the user dynamically gets or loses access to accounts and applications that you assigned to the group.

- Configure permissions based on job functions, see [Create a permission sets](#).

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in IAM Identity Center and can be provisioned to one or more AWS accounts. You can assign more than one permission set to a user.

Note

As an IAM Identity Center administrator, you'll occasionally need to replace older IdP certificates with newer ones. For example, you might need to replace an IdP certificate when the expiration date on the certificate approaches. The process of replacing an older certificate with a newer one is referred to as certificate rotation. Make sure to review how to [manage the SAML certificates](#) for Google Workspace.

Using IAM Identity Center to connect with your JumpCloud Directory Platform

IAM Identity Center supports automatic provisioning (synchronization) of user information from JumpCloud Directory Platform into IAM Identity Center. This provisioning uses the [Security Assertion Markup Language \(SAML\) 2.0](#) protocol. You configure this connection in JumpCloud using your IAM Identity Center SCIM endpoint and access token. When you configure SCIM synchronization, you create a mapping of your user attributes in JumpCloud to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and JumpCloud.

This guide is based on JumpCloud as of June 2021. Steps for newer versions may vary. This guide contains a few notes regarding configuration of user authentication through SAML.

The following steps walk you through how to enable automatic provisioning of users and groups from JumpCloud to IAM Identity Center using the SCIM protocol.

Note

Before you begin deploying SCIM, we recommend that you first review the [Considerations for using automatic provisioning](#). Then continue reviewing additional considerations in the next section.

Topics

- [Prerequisites](#)
- [SCIM considerations](#)
- [Step 1: Enable provisioning in IAM Identity Center](#)

- [Step 2: Configure provisioning in JumpCloud](#)
- [\(Optional\) Step 3: Configure user attributes in JumpCloud for access control in IAM Identity Center](#)
- [\(Optional\) Passing attributes for access control](#)

Prerequisites

You will need the following before you can get started:

- JumpCloud subscription or free trial. To sign up for a free trial visit [JumpCloud](#).
- An IAM Identity Center enabled account ([free](#)). For more information, see [Enable IAM Identity Center](#).
- A SAML connection from your JumpCloud account to IAM Identity Center, as described in [JumpCloud documentation for IAM Identity Center](#).
- Associate the IAM Identity Center connector with the groups you want to allow access to AWS accounts.

SCIM considerations

The following are considerations when using JumpCloud federation for IAM Identity Center.

- Only groups associated with the AWS Single Sign-On connector in JumpCloud will be synchronized with SCIM.
- Only one phone number attribute can be synchronized and the default is "work phone."
- Users in JumpCloud directory must have first and last names configured to be synchronized to IAM Identity Center with SCIM.
- Attributes are still synchronized if the user is disabled in IAM Identity Center but still activate in JumpCloud.
- You can choose to enable SCIM sync for only user information by unchecking the "Enable management of User Groups and Group membership" in the connector.

Step 1: Enable provisioning in IAM Identity Center

In this first step, you use the IAM Identity Center console to enable automatic provisioning.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-5555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

Now that you have set up provisioning in the IAM Identity Center console, you need to complete the remaining tasks using the JumpCloud IAM Identity Center connector. These steps are described in the following procedure.

Step 2: Configure provisioning in JumpCloud

Use the following procedure in the JumpCloud IAM Identity Center connector to enable provisioning with IAM Identity Center. This procedure assumes that you have already added the JumpCloud IAM Identity Center connector to your JumpCloud admin portal and groups. If you have not yet done so, refer to [Prerequisites](#), and then complete this procedure to configure SCIM provisioning.

To configure provisioning in JumpCloud

1. Open the JumpCloud IAM Identity Center connector that you installed as part of configuring SAML for JumpCloud (**User Authentication > IAM Identity Center**). See [Prerequisites](#).
2. Choose the **IAM Identity Center** connector, and then choose the third tab **Identity Management**.
3. Check the box for **Enable management of User Groups and Group membership in this application** if you want groups to SCIM sync.
4. Click on **Configure**.
5. In the previous procedure, you copied the **SCIM endpoint** value in IAM Identity Center. Paste that value into the **Base URL** field in the JumpCloud IAM Identity Center connector.
6. From the previous procedure you copied the **Access token** value in IAM Identity Center. Paste that value into the **Token Key** field in the JumpCloud IAM Identity Center connector.
7. Click **Activate** to apply the configuration.
8. Make sure you have a green indicator next to **Single Sign-On activated**.
9. Move to the fourth tab **User Groups** and check the groups you want to be provisioned with SCIM.
10. Click **Save** at the bottom once you are done.
11. To verify that users have been successfully synchronized to IAM Identity Center, return to the IAM Identity Center console and choose **Users**. Synchronized users from JumpCloud appear on the **Users** page. These users can now be assigned to accounts within IAM Identity Center.

(Optional) Step 3: Configure user attributes in JumpCloud for access control in IAM Identity Center

This is an optional procedure for JumpCloud should you choose to configure attributes for IAM Identity Center to manage access to your AWS resources. The attributes that you define in JumpCloud are passed in a SAML assertion to IAM Identity Center. You then create a permission set in IAM Identity Center to manage access based on the attributes you passed from JumpCloud.

Before you begin this procedure, you must first enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

To configure user attributes in JumpCloud for access control in IAM Identity Center

1. Open the JumpCloud IAM Identity Center connector that you installed as part of configuring SAML for JumpCloud (**User Authentication > IAM Identity Center**).
2. Choose the **IAM Identity Center** connector. Then, choose the second tab **IAM Identity Center**.
3. At the bottom of this tab you have **User Attribute Mapping**, choose **Add new attribute**, and then do the following: You must perform these steps for each attribute you will add for use in IAM Identity Center for access control.
 - a. In the **Service Provide Attribute Name** field, enter `https://aws.amazon.com/SAML/Attributes/AccessControl:AttributeName`. Replace **AttributeName** with the name of the attribute you are expecting in IAM Identity Center. For example, `https://aws.amazon.com/SAML/Attributes/AccessControl:Email`.
 - b. In the **JumpCloud Attribute Name** field, choose user attributes from your JumpCloud directory. For example, **Email (Work)**.
4. Choose **Save**.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Configure SAML and SCIM with Microsoft Entra ID and IAM Identity Center

AWS IAM Identity Center supports integration with [Security Assertion Markup Language \(SAML\) 2.0](#) as well as [automatic provisioning](#) (synchronization) of user and group information from Microsoft Entra ID (formerly known as Azure Active Directory or Azure AD) into IAM Identity Center using the [System for Cross-domain Identity Management \(SCIM\) 2.0](#) protocol.

Objective

In this tutorial, you will set up a test lab and configure a SAML connection and SCIM provisioning between Microsoft Entra ID and IAM Identity Center. During the initial preparation steps, you'll create a test user (Nikki Wolf) in both Microsoft Entra ID and IAM Identity Center which you'll use to test the SAML connection in both directions. Later, as part of the SCIM steps, you'll create a different test user (Richard Roe) to verify that new attributes in Microsoft Entra ID are synchronizing to IAM Identity Center as expected.

Prerequisites

Before you can get started with this tutorial, you'll first need to set up the following:

- A Microsoft Entra ID tenant. For more information, see [Quickstart: Set up a tenant](#) in Microsoft documentation.
- An AWS IAM Identity Center-enabled account. For more information, see [Enable IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

Considerations

The following are important considerations about Microsoft Entra ID that can affect how you plan to implement [automatic provisioning](#) with IAM Identity Center in your production environment using the SCIM v2 protocol.

Automatic Provisioning

Before you begin deploying SCIM, we recommend that you first review [Considerations for using automatic provisioning](#).

Attributes for access control

Attributes for access control is used in permission policies that determine who in your identity source can access your AWS resources. If an attribute is removed from a user in Microsoft Entra ID, that attribute will not be removed from the corresponding user in IAM Identity Center. This is a known limitation in Microsoft Entra ID. If an attribute is changed to a different (non-empty) value on a user, that change will be synchronized to IAM Identity Center.

Nested Groups

The Microsoft Entra ID user provisioning service can't read or provision users in nested groups. Only users that are immediate members of an explicitly assigned group can be read and provisioned. Microsoft Entra ID doesn't recursively unpack the group memberships of indirectly assigned users or groups (users or groups that are members of a group that is directly assigned). For more information, see [Assignment-based scoping](#) in the Microsoft documentation. Alternatively, you can use [IAM Identity Center ID AD Sync](#) to integrate Active Directory groups with IAM Identity Center.

Dynamic Groups

The Microsoft Entra ID user provisioning service can read and provision users in [dynamic groups](#). See below for an example showing the users and groups structure while using dynamic groups and how they are displayed in IAM Identity Center. These users and groups were provisioned from Microsoft Entra ID into IAM Identity Center via SCIM

For example, if Microsoft Entra ID structure for dynamic groups is as follows:

1. Group A with members ua1, ua2
2. Group B with members ub1
3. Group C with members uc1
4. Group K with a rule to include members of Group A, B, C
5. Group L with a rule to include members Group B and C

After the user and group information is provisioned from Microsoft Entra ID into IAM Identity Center through SCIM, the structure will be as follows:

1. Group A with members ua1, ua2
2. Group B with members ub1
3. Group C with members uc1
4. Group K with members ua1, ua2, ub1, uc1

5. Group L with members ub1, uc1

When you configure automatic provisioning using dynamic groups, keep the following considerations in mind.

- A dynamic group can include a nested group. However, Microsoft Entra ID provisioning service doesn't flatten the nested group. For example, if you have the following Microsoft Entra ID structure for dynamic groups:
 - Group A is a parent of group B.
 - Group A has ua1 as a member.
 - Group B has ub1 as a member.

The dynamic group that includes Group A will only include the direct members of group A (that is, ua1). It won't recursively include members of group B.

- Dynamic groups can't contain other dynamic groups. For more information, see [Preview limitations](#) in the Microsoft documentation.

Step 1: Prepare your Microsoft tenant

In this step, you will walk through how to install and configure your AWS IAM Identity Center enterprise application and assign access to a newly created Microsoft Entra ID test user.

Step 1.1 >

Step 1.1: Set up the AWS IAM Identity Center enterprise application in Microsoft Entra ID

In this procedure, you install the AWS IAM Identity Center enterprise application in Microsoft Entra ID. You will need this application later to configure your SAML connection with AWS.

1. Sign in to the [Microsoft Entra admin center](#) as at least a Cloud Application Administrator.
2. Navigate to **Identity > Applications > Enterprise applications**, and then choose **New application**.
3. On the **Browse Microsoft Entra Gallery** page, enter **AWS IAM Identity Center** in the search box.
4. Select **AWS IAM Identity Center** from the results.
5. Choose **Create**.

Step 1.2 >

Step 1.2: Create a test user in Microsoft Entra ID

Nikki Wolf is the name of your Microsoft Entra ID test user that you will create in this procedure.

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Users > All users**.
2. Select **New user**, and then choose **Create new user** at the top of the screen.
3. In **User principal name**, enter **NikkiWolf**, and then select your preferred domain and extension. For example, *NikkiWolf@example.org*.
4. In **Display name**, enter **NikkiWolf**.
5. In **Password**, enter a strong password or select the eye icon to show the password that was auto-generated, and either copy or write down the value that's displayed.
6. Choose **Properties**, in **First name**, enter **Nikki**. In **Last name**, enter **Wolf**.
7. Choose **Review + create**, and then choose **Create**.

Step 1.3

Step 1.3: Test Nikki's experience prior to assigning her permissions to AWS IAM Identity Center

In this procedure, you will verify what Nikki can successfully sign into her Microsoft [My Account portal](#).

1. In the same browser, open a new tab, go to the [My Account portal](#) sign-in page, and enter Nikki's full email address. For example, *NikkiWolf@example.org*.
2. When prompted, enter Nikki's password, and then choose **Sign in**. If this was an auto-generated password, you will be prompted to change the password.
3. On the **Action Required** page, choose **Ask later** to bypass the prompt for additional security methods.
4. On the **My account** page, in the left navigation pane, choose **My Apps**. Notice that besides **Add-ins**, no apps are displayed at this time. You'll add an **AWS IAM Identity Center** app that will appear here in a later step.

Step 1.4

Step 1.4: Assign permissions to Nikki in Microsoft Entra ID

Now that you have verified that Nikki can successfully access the **My account portal**, use this procedure to assign her user to the **AWS IAM Identity Center** app.

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Applications > Enterprise applications** and then choose **AWS IAM Identity Center** from the list.
2. On the left, choose **Users and groups**.
3. Choose **Add user/group**. You can ignore the message stating that groups are not available for assignment. This tutorial does not use groups for assignments.
4. On the **Add Assignment** page, under **Users**, choose **None Selected**.
5. Select **NikkiWolf**, and then choose **Select**.
6. On the **Add Assignment** page, choose **Assign**. NikkiWolf now appears in the list of users who are assigned to the **AWS IAM Identity Center** app.

Step 2: Prepare your AWS account

In this step, you'll walk through how to use **IAM Identity Center** to configure access permissions (via permission set), manually create a corresponding Nikki Wolf user, and assign her the necessary permissions to administer resources in AWS.

Step 2.1 >

Step 2.1: Create a RegionalAdmin permission set in IAM Identity Center

This permission set will be used to grant Nikki the necessary AWS account permissions required to manage Regions from the **Account** page within the AWS Management Console. All other permissions to view or manage any other information for Nikki's account is denied by default.

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Choose **Create permission set**.
4. On the **Select permission set type** page, select **Custom permission set**, and then choose **Next**.
5. Select **Inline policy** to expand it, and then create a policy for the permission set using the following steps:
 - a. Choose **Add new statement** to create a policy statement.

- b. Under **Edit statement**, select **Account** from the list, and then choose the following checkboxes.
 - **ListRegions**
 - **GetRegionOptStatus**
 - **DisableRegion**
 - **EnableRegion**
- c. Next to **Add a resource**, choose **Add**.
- d. On the **Add resource** page, under **Resource type**, select **All Resources**, and then choose **Add resource**. Verify that your policy looks like the following:

```
{
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "account:ListRegions",
        "account:DisableRegion",
        "account:EnableRegion",
        "account:GetRegionOptStatus"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

6. Choose **Next**.
7. On the **Specify permission set details** page, under **Permission set name**, enter **RegionalAdmin**, and then choose **Next**.
8. On the **Review and create** page, choose **Create**. You should see **RegionalAdmin** displayed in the list of permission sets.

Step 2.2 >

Step 2.2: Create a corresponding NikkiWolf user in IAM Identity Center

Since the SAML protocol does not provide a mechanism to query the IdP (Microsoft Entra ID) and automatically create users here in IAM Identity Center, use the following procedure to manually create a user in IAM Identity Center that mirrors the core attributes from Nikki Wolfs user in Microsoft Entra ID.

1. Open the [IAM Identity Center console](#).
2. Choose **Users**, choose **Add user**, and then provide the following information:
 - a. For both **Username** and **Email address** – Enter the same **NikkiWolf@yourcompanydomain.extension** that you used when creating your Microsoft Entra ID user. For example, *NikkiWolf@example.org*.
 - b. **Confirm email address** – Re-enter the email address from the previous step
 - c. **First name** – Enter **Nikki**
 - d. **Last name** – Enter **Wolf**
 - e. **Display name** – Enter **Nikki Wolf**
3. Choose **Next** twice, then choose **Add user**.
4. Select **Close**.

Step 2.3

Step 2.3: Assign Nikki to the RegionalAdmin permission set in IAM Identity Center

Here you locate the AWS account in which Nikki will administer Regions, and then assign the necessary permissions required for her to successfully access the AWS access portal.

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **AWS accounts**.
3. Select the checkbox next to the account name (for example, *Sandbox*) where you want to grant Nikki access to manage Regions, and then choose **Assign users and groups**.
4. On the **Assign users and groups** page, choose the **Users** tab, find and check the box next to Nikki, and then choose **Next**.

Step 3: Configure and test your SAML connection

In this step, you configure your SAML connection using the AWS IAM Identity Center enterprise application in Microsoft Entra ID together with the external IdP settings in IAM Identity Center.

Step 3.1 >

Step 3.1: Collect required service provider metadata from IAM Identity Center

In this step, you will launch the **Change identity source** wizard from within the IAM Identity Center console and retrieve the metadata file and the AWS specific sign-in URL you'll need to enter when configuring the connection with Microsoft Entra ID in the next step.

1. In the [IAM Identity Center console](#), choose **Settings**.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Change identity source**.
3. On the **Choose identity source** page, select **External identity provider**, and then choose **Next**.
4. On the **Configure external identity provider** page, under **Service provider metadata**, choose **Download metadata file** to download the XML file.
5. In the same section, locate the **AWS access portal sign-in URL** value and copy it. You will need to enter this value when prompted in the next step.
6. Leave this page open, and move to the next step (**Step 3.2**) to configure the AWS IAM Identity Center enterprise application in Microsoft Entra ID. Later, you'll return to this page to complete the process.

Step 3.2 >

Step 3.2: Configure the AWS IAM Identity Center enterprise application in Microsoft Entra ID

This procedure establishes one-half of the SAML connection on the Microsoft side using the values from the metadata file and Sign-On URL you obtained in the last step.

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Applications > Enterprise applications** and then choose **AWS IAM Identity Center**.
2. On the left, choose **2. Set up Single sign-on**.
3. On the **Set up Single Sign-On with SAML** page, choose **SAML**. Then choose **Upload metadata file**, choose the folder icon, select the service provider metadata file that you downloaded in the previous step, and then choose **Add**.

4. On the **Basic SAML Configuration** page, verify that both the **Identifier** and **Reply URL** values now point to endpoints in AWS that start with `https://<REGION>.signin.aws.amazon.com/platform/saml/`.
5. Under **Sign on URL (Optional)**, paste in the **AWS access portal sign-in URL** value you copied in the previous step (**Step 3.1**), choose **Save**, and then choose **X** to close the window.
6. If prompted to test single sign-on with AWS IAM Identity Center, choose **No I'll test later**. You will do this verification in a later step.
7. On the **Set up Single Sign-On with SAML** page, in the **SAML Certificates** section, next to **Federation Metadata XML**, choose **Download** to save the metadata file to your system. You will need to upload this file when prompted in the next step.

Step 3.3 >

Step 3.3: Configure the Microsoft Entra ID external IdP in AWS IAM Identity Center

Here you will return to the **Change identity source** wizard in the IAM Identity Center console to complete the second-half of the SAML connection in AWS.

1. Return to the browser session you left open from **Step 3.1** in the IAM Identity Center console.
2. On the **Configure external identity provider** page, in the **Identity provider metadata** section, under **IdP SAML metadata**, choose the **Choose file** button, and select the identity provider metadata file that you downloaded from Microsoft Entra ID in the previous step, and then choose **Open**.
3. Choose **Next**.
4. After you read the disclaimer and are ready to proceed, enter **ACCEPT**.
5. Choose **Change identity source** to apply your changes.

Step 3.4 >

Step 3.4: Test that Nikki is redirected to the AWS access portal

In this procedure, you will test the SAML connection by signing in to Microsoft's **My Account portal** with Nikki's credentials. Once authenticated, you'll select the AWS IAM Identity Center application which will redirect Nikki to the AWS access portal.

1. Go to the [My Account portal](#) sign in page, and enter Nikki's full email address. For example, **NikkiWolf@example.org**.
2. When prompted, enter Nikki's password, and then choose **Sign in**.
3. On the **My account** page, in the left navigation pane, choose **My Apps**.
4. On the **My Apps** page, select the app named **AWS IAM Identity Center**. This should prompt you for additional authentication.
5. On Microsoft's sign in page, choose your NikkiWolf credentials. If prompted a second time for authentication, choose your NikkiWolf credentials again. This should automatically redirect you to the AWS access portal.

 **Tip**

If you are not redirected successfully, check to make sure the **AWS access portal sign-in URL** value you entered in **Step 3.2** matches the value you copied from **Step 3.1**.

6. Verify that your AWS accounts display.

 **Tip**

If the page is empty and no AWS accounts display, confirm that Nikki was successfully assigned to the **RegionalAdmin** permission set (see **Step 2.3**).

Step 3.5

Step 3.5: Test Nikki's level of access to manage her AWS account

In this step, you will check to determine Nikki's level of access to manage the Region settings for her AWS account. Nikki should only have sufficient administrator privileges to manage Regions from the **Accounts** page.

1. In the AWS access portal, choose the **Accounts** tab to display the list of accounts. The account names, account IDs, and email addresses associated with any accounts where you've defined permission sets appear.

2. Choose the account name (for example, *Sandbox*) where you applied the permission set (see **Step 2.3**). This will expand the list of permission sets that Nikki can choose from to manage her account.
3. Next to **RegionalAdmin** choose **Management console** to assume the role you defined in the **RegionalAdmin** permission set. This will redirect you to the AWS Management Console home page.
4. In the upper-right corner of the console, choose your account name, and then choose **Account**. This will take you to the **Account** page. Notice that all other sections on this page display a message that you don't have the necessary permissions to view or modify those settings.
5. On the **Account** page, scroll down to the section **AWS Regions**. Select a checkbox for any available Region in the table. Notice that Nikki does have the necessary permissions to **Enable** or **Disable** the list of Regions for her account as was intended.

Nicely done!

Steps 1 through 3 helped you to successfully implement and test your SAML connection. Now, to complete the tutorial, we encourage you to move on to Step 4 to implement automatic provisioning.

Step 4: Configure and test your SCIM synchronization

In this step, you will set up [automatic provisioning](#) (synchronization) of user information from Microsoft Entra ID into IAM Identity Center using the SCIM v2.0 protocol. You configure this connection in Microsoft Entra ID using your SCIM endpoint for IAM Identity Center and a bearer token that is created automatically by IAM Identity Center.

When you configure SCIM synchronization, you create a mapping of your user attributes in Microsoft Entra ID to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and Microsoft Entra ID.

The following steps walk you through how to enable automatic provisioning of users that primarily reside in Microsoft Entra ID to IAM Identity Center using the IAM Identity Center app in Microsoft Entra ID.

Step 4.1 >

Step 4.1: Create a second test user in Microsoft Entra ID

For testing purposes, you will create a new user (Richard Roe) in Microsoft Entra ID. Later, after you set up SCIM synchronization, you will test that this user and all relevant attributes were synced successfully to IAM Identity Center.

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Users > All users**.
2. Select **New user**, and then choose **Create new user** at the top of the screen.
3. In **User principal name**, enter **RichRoe**, and then select your preferred domain and extension. For example, *RichRoe@example.org*.
4. In **Display name**, enter **RichRoe**.
5. In **Password**, enter a strong password or select the eye icon to show the password that was auto-generated, and either copy or write down the value that's displayed.
6. Choose **Properties**, and then provide the following values:
 - **First name** - Enter **Richard**
 - **Last name** - Enter **Roe**
 - **Job title** - Enter **Marketing Lead**
 - **Department** - Enter **Sales**
 - **Employee ID** - Enter **12345**
7. Choose **Review + create**, and then choose **Create**.


Step 4.2 >

Step 4.2: Enable automatic provisioning in IAM Identity Center

In this procedure, you will use the IAM Identity Center console to enable automatic provisioning of users and groups coming from Microsoft Entra ID into IAM Identity Center.

1. Open the [IAM Identity Center console](#), and choose **Settings** in the left navigation pane.
2. On the **Settings** page, under the **Identity source** tab, notice that **Provisioning method** is set to **Manual**.
3. Locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.

4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in the next step when you configure provisioning in Microsoft Entra ID.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

 **Warning**

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward.

5. Choose **Close**.
6. Under the **Identity source** tab, notice that **Provisioning method** is now set to **SCIM**.

Step 4.3 >

Step 4.3: Configure automatic provisioning in Microsoft Entra ID

Now that you have your RichRoe test user in place and have enabled SCIM in IAM Identity Center, you can proceed with configuring the SCIM synchronization settings in Microsoft Entra ID.

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Applications > Enterprise applications** and then choose **AWS IAM Identity Center**.
2. Choose **Provisioning**, under **Manage**, choose **Provisioning** again.
3. In **Provisioning Mode** select **Automatic**.
4. Under **Admin Credentials**, in **Tenant URL** paste in the **SCIM endpoint** URL value you copied earlier in **Step 4.1**. In **Secret Token**, paste in the **Access token** value.
5. Choose **Test Connection**. You should see a message indicating that the tested credentials were successfully authorized to enable provisioning.
6. Choose **Save**.
7. Under **Manage**, choose **Users and groups**, and then choose **Add user/group**.
8. On the **Add Assignment** page, under **Users**, choose **None Selected**.

9. Select **RichRoe**, and then choose **Select**.
10. On the **Add Assignment** page, choose **Assign**.
11. Choose **Overview**, and then choose **Start provisioning**.

Step 4.4

Step 4.4: Verify that synchronization occurred

In this section, you will verify that Richard's user was successfully provisioned and that all attributes are displayed in IAM Identity Center.

1. In the [IAM Identity Center console](#), choose **Users**.
2. On the **Users** page, you should see your **RichRoe** user displayed. Notice that in the **Created by** column the value is set to **SCIM**.
3. Choose **RichRoe**, under **Profile**, verify that the following attributes were copied from Microsoft Entra ID.
 - **First name - Richard**
 - **Last name - Roe**
 - **Department - Sales**
 - **Title - Marketing Lead**
 - **Employee number - 12345**

Now that Richard's user has been created in IAM Identity Center, you can assign it to any permission set so you can control the level of access he has to your AWS resources. For example, you could assign **RichRoe** to the **RegionalAdmin** permission set you used earlier to grant Nikki the permissions to manage Regions (see **Step 2.3**) and then test his level of access using **Step 3.5**.

Congratulations!

You have successfully set up a SAML connection between Microsoft and AWS and have verified that automatic provisioning is working to keep everything in sync. Now you can apply what you've learned to more smoothly set up your production environment.

Troubleshooting SCIM issues with Microsoft Entra ID

If you are experiencing issues with Microsoft Entra ID users not synchronizing to IAM Identity Center, it might be due to a syntax issue that IAM Identity Center has flagged when a new user is being added to IAM Identity Center. You can confirm this by checking the Microsoft Entra ID audit logs for failed events, such as an 'Export'. The **Status Reason** for this event will state:

```
{"schema":["urn:ietf:params:scim:api:messages:2.0:Error"],"detail":"Request is unparsable, syntactically incorrect, or violates schema.","status":"400"}
```

You can also check AWS CloudTrail for the failed event. This can be done by searching in the **Event History** console of CloudTrail using the following filter:

```
"eventName":"CreateUser"
```

The error in the CloudTrail event will state the following:

```
"errorCode": "ValidationException",  
  "errorMessage": "Currently list attributes only allow single item"
```

Ultimately, this exception means that one of the values passed from Microsoft Entra ID contained more values than anticipated. The solution here is to review the attributes of the user in Microsoft Entra ID, ensuring that none contain duplicate values. One common example of duplicate values is having multiple values present for contact numbers such as **mobile**, **work**, and **fax**. Although separate values, they are all passed to IAM Identity Center under the single parent attribute **phoneNumbers**.

For general SCIM troubleshooting tips, see [Troubleshooting IAM Identity Center issues](#).

Step 5: (Optional) Configure ABAC

Now that you have successfully configured SAML and SCIM, you can optionally choose to configure attribute-based access control (ABAC). ABAC is an authorization strategy that defines permissions based on attributes.

With Microsoft Entra ID, you can use either of the following two methods to configure ABAC for use with IAM Identity Center.

Configure user attributes in Microsoft Entra ID for access control in IAM Identity Center

Configure user attributes in Microsoft Entra ID for access control in IAM Identity Center

In the following procedure, you will determine which attributes in Microsoft Entra ID should be used by IAM Identity Center to manage access to your AWS resources. Once defined, Microsoft Entra ID sends these attributes to IAM Identity Center through SAML assertions. You will then need to [Create a permission set](#) in IAM Identity Center to manage access based on the attributes you passed from Microsoft Entra ID.

Before you begin this procedure, you first need to enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

1. In the [Microsoft Entra admin center](#) console, navigate to **Identity > Applications > Enterprise applications** and then choose **AWS IAM Identity Center**.
2. Choose **Single sign-on**.
3. In the **Attributes & Claims** section, choose **Edit**.
4. On the **Attributes & Claims** page, do the following:
 - a. Choose **Add new claim**
 - b. For **Name**, enter `AccessControl:AttributeName`. Replace *AttributeName* with the name of the attribute you are expecting in IAM Identity Center. For example, `AccessControl:Department`.
 - c. For **Namespace**, enter `https://aws.amazon.com/SAML/Attributes`.
 - d. For **Source**, choose **Attribute**.
 - e. For **Source attribute**, use the drop-down list to choose the Microsoft Entra ID user attributes. For example, `user.department`.
5. Repeat the previous step for each attribute you need to send to IAM Identity Center in the SAML assertion.
6. Choose **Save**.

Configure ABAC using IAM Identity Center

Configure ABAC using IAM Identity Center

With this method, you use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. You can use this element to pass attributes

as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `Department=billing`, use the following attribute:

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/
AccessControl:Department">
<saml:AttributeValue>billing
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Configure SAML and SCIM with Okta and IAM Identity Center

You can automatically provision or synchronize user and group information from Okta into IAM Identity Center using the [System for Cross-domain Identity Management \(SCIM\) 2.0 protocol](#). To configure this connection in Okta, you use your SCIM endpoint for IAM Identity Center and a bearer token that is created automatically by IAM Identity Center. When you configure SCIM synchronization, you create a mapping of your user attributes in Okta to the named attributes in IAM Identity Center. This mapping matches the expected user attributes between IAM Identity Center and your Okta account.

Okta supports the following provisioning features when connected to IAM Identity Center through SCIM:

- Create users – Users assigned to the IAM Identity Center application in Okta are provisioned in IAM Identity Center.
- Update user attributes – Attribute changes for users who are assigned to the IAM Identity Center application in Okta are updated in IAM Identity Center.
- Deactivate users – Users who are unassigned from the IAM Identity Center application in Okta are disabled in IAM Identity Center.
- Group push – Groups (and their members) in Okta are synchronized to IAM Identity Center.

Note

To minimize administrative overhead in both Okta and IAM Identity Center, we recommend that you assign and *push* groups instead of individual users.

Objective

In this tutorial, you will walk through setting up a SAML connection with Okta IAM Identity Center. Later, you will synchronize users from Okta, using SCIM. In this scenario, you manage all users and groups in Okta. Users sign in through the Okta portal. To verify everything is configured correctly, after completing the configuration steps you will sign in as an Okta user and verify access to AWS resources.

Note

You can sign up for an Okta account ([free trial](#)) that has Okta's [IAM Identity Center application](#) installed. For paid Okta products, you might need to confirm that your Okta license supports *lifecycle management* or similar capabilities that enable outbound provisioning. These features might be necessary to configure SCIM from Okta to IAM Identity Center.

If you haven't enabled IAM Identity Center yet, see [Enabling AWS IAM Identity Center](#).

Considerations

- Before you configure SCIM provisioning between Okta and IAM Identity Center, we recommend that you first review [Considerations for using automatic provisioning](#).
- Every Okta user must have a **First name**, **Last name**, **Username** and **Display name** value specified.
- Each Okta user has only a single value per data attribute, such as email address or phone number. Any users that have multiple values will fail to synchronize. If there are users that have multiple values in their attributes, remove the duplicate attributes before attempting to provision the user in IAM Identity Center. For example, only one phone number attribute can be synchronized, since the default phone number attribute is "work phone", use the "work phone" attribute to store the user's phone number, even if the phone number for the user is a home phone or a mobile phone.

- When using Okta with IAM Identity Center, IAM Identity Center is generally configured as an Application in Okta. This allows you to configure multiple instances of IAM Identity Center as multiple applications, supporting access to multiple AWS Organizations, within a single instance of the Okta.
- Entitlements and role attributes aren't supported and can't be synchronized with IAM Identity Center.
- Using the same Okta group for both assignments and group push isn't currently supported. To maintain consistent group memberships between Okta and IAM Identity Center, create a separate group and configure it to push groups to IAM Identity Center.

Step 1: Okta: Obtain the SAML metadata from your Okta account

1. Sign in to the Okta admin dashboard, expand **Applications**, then select **Applications**.
2. On the **Applications** page, choose **Browse App Catalog**.
3. In the search box, type **AWS IAM Identity Center**, select the app to add the IAM Identity Center app.
4. Select the **Sign On** tab.
5. Under **SAML Signing Certificates**, select **Actions**, and then select **View IdP Metadata**. A new browser tab opens showing the document tree of an XML file. Select all of the XML from `<md:EntityDescriptor>` to `</md:EntityDescriptor>` and copy it to a text file.
6. Save the text file as `metadata.xml`.

Leave the Okta admin dashboard open, you will continue using this console in the later steps.

Step 2: IAM Identity Center: Configure Okta as the identity source for IAM Identity Center

1. Open the [IAM Identity Center console](#) as a user with administrative privileges.
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, choose **Actions**, and then choose **Change identity source**.
4. Under **Choose identity source**, select **External identity provider**, and then choose **Next**.
5. Under **Configure external identity provider**, do the following:

- a. Under **Service provider metadata**, choose **Download metadata file** to download the IAM Identity Center metadata file and save it on your system. You will provide the IAM Identity Center SAML metadata file to Okta later in this tutorial.

Copy the following items to a text file for easy access:

- **IAM Identity Center Assertion Consumer Service (ACS) URL**
- **IAM Identity Center issuer URL**

You'll need these values later in this tutorial.

- b. Under **Identity provider metadata**, under **IdP SAML metadata**, select **Choose file** and then select the metadata .xml file you created in the previous step.
 - c. Choose **Next**.
6. After you read the disclaimer and are ready to proceed, enter **ACCEPT**.
 7. Choose **Change identity source**.

Leave the AWS console open, you will continue using this console in the next step.

8. Return to the Okta admin dashboard and select the **Sign On** tab of the AWS IAM Identity Center app, then select **Edit**.
9. Under **Advanced Sign-on Settings** enter the following:
 - For **ACS URL**, enter the value you copied for **IAM Identity Center Assertion Consumer Service (ACS) URL**
 - For **Issuer URL**, enter the value you copied for **IAM Identity Center issuer URL**
 - For **Application username format**, select one of the options from the menu.

Ensure the value you choose is unique for each user. For this tutorial, select **Okta username**

10. Choose **Save**.

You are now ready to provision users from Okta to IAM Identity Center. Leave the Okta admin dashboard open, and return to the IAM Identity Center console for the next step.

Step 3: IAM Identity Center and Okta: Provision Okta users

1. In the IAM Identity Center console on the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
2. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options:
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

3. Choose **Close**.
4. Return to the Okta admin dashboard and navigate to the IAM Identity Center app.
5. On the **IAM Identity Center app** page, choose the **Provisioning** tab, and then in the left navigation under **Settings**, choose **Integration**.
6. Choose **Edit**, and then select the checkbox next to **Enable API integration** to enable automatic provisioning.
7. Configure Okta with the SCIM provisioning values from AWS IAM Identity Center that you copied earlier in this step:
 - a. In the **Base URL** field, enter the **SCIM endpoint** value.
 - b. In the **API Token** field, enter the **Access token** value.
8. Choose **Test API Credentials** to verify the credentials entered are valid.

The message **AWS IAM Identity Center was verified successfully!** displays.

9. Choose **Save**. You're moved to the **Settings** section, with **Integration** selected.
10. Under **Settings**, choose **To App**, and then select the **Enable** checkbox for each of the **Provisioning to App** features you want to enable. For this tutorial, select all the options.

11. Choose **Save**.

You are now ready to synchronize your users from Okta with IAM Identity Center.

Step 4: Okta: Synchronize users from Okta with IAM Identity Center

By default, no groups or users are assigned to your Okta IAM Identity Center app. Provisioning groups provisions the users that are members of the group. Complete the following steps to synchronize groups and users with AWS IAM Identity Center.

1. In the **Okta IAM Identity Center app** page, choose the **Assignments** tab. You can assign both people and groups to the IAM Identity Center app.

a. To assign people:

- In the **Assignments** page, choose **Assign**, and then choose **Assign to people**.
- Choose the Okta users that you want to have access to the IAM Identity Center app. Choose **Assign**, choose **Save and Go Back**, and then choose **Done**.

This starts the process of provisioning the users into IAM Identity Center.

b. To assign groups:

- In the **Assignments** page, choose **Assign**, and then choose **Assign to groups**.
- Choose the Okta groups that you want to have access to the IAM Identity Center app. Choose **Assign**, choose **Save and Go Back**, and then choose **Done**.

This starts the process of provisioning the users in the group into IAM Identity Center.

Note

You might be required to specify additional attributes for the group if they aren't present in all of the user records. The attributes specified for the group will override any individual attribute values.

2. Choose the **Push Groups** tab. Choose the Okta group you want to synchronize with IAM Identity Center. Choose **Save**.

The group status changes to **Active** after the group and its members have been pushed to IAM Identity Center.

3. Return to the **Assignments** tab.
4. To add individual Okta users to IAM Identity Center, use the following steps:
 - a. In the **Assignments** page, choose **Assign**, and then choose **Assign to People**.
 - b. Choose the Okta users that you want to have access to the IAM Identity Center app. Choose **Assign**, choose **Save and Go Back**, and then choose **Done**.

This starts the process of provisioning the individual users into IAM Identity Center.

 **Note**

You can also assign users and groups to the AWS IAM Identity Center app, from the **Applications** page of the Okta admin dashboard. To do this select the **Settings** icon and then choose **Assign to Users** or **Assign to Groups** and then specify the user or group.

5. Return to the IAM Identity Center console. In the left navigation, select **Users**, you should see the user list populated by your Okta users.

 **Congratulations!**

You have successfully set up a SAML connection between Okta and AWS and have verified that automatic provisioning is working. You can now assign these users to accounts and applications in **IAM Identity Center**. For this tutorial, in the next step let's designate one of the users as the IAM Identity Center administrator by granting them administrative permissions to the management account.

Step 5: IAM Identity Center: Grant Okta users access to accounts

1. In the IAM Identity Center navigation pane, under **Multi-account permissions**, choose **AWS accounts**.

2. On the **AWS accounts** page the **Organizational structure** displays your organizational root with your accounts underneath it in the hierarchy. Select the checkbox for your management account, then select **Assign users or groups**.
3. The **Assign users and groups** workflow displays. It consists of three steps:
 - a. For **Step 1: Select users and groups**, choose the user that will be performing the administrator job function. Then choose **Next**.
 - b. For **Step 2: Select permission sets**, choose **Create permission set** to open a new tab that walks you through the three sub-steps involved in creating a permission set.
 - i. For **Step 1: Select permission set type** complete the following:
 - In **Permission set type**, choose **Predefined permission set**.
 - In **Policy for predefined permission set**, choose **AdministratorAccess**.

Choose **Next**.

- ii. For **Step 2: Specify permission set details**, keep the default settings, and choose **Next**.

The default settings create a permission set named *AdministratorAccess* with session duration set to one hour.

- iii. For **Step 3: Review and create**, verify that the **Permission set type** uses the AWS managed policy **AdministratorAccess**. Choose **Create**. On the **Permission sets** page, a notification appears informing you that the permission set was created. You can close this tab in your web browser now.

On the **Assign users and groups** browser tab, you are still on **Step 2: Select permission sets** from which you started the create permission set workflow.

In the **Permissions sets** area, choose the **Refresh** button. The *AdministratorAccess* permission set you created appears in the list. Select the checkbox for that permission set and then choose **Next**.

- c. For **Step 3: Review and submit**, review the selected user and permission set, then choose **Submit**.

The page updates with a message that your AWS account is being configured. Wait until the process completes.

You are returned to the AWS accounts page. A notification message informs you that your AWS account has been reprovisioned and the updated permission set applied. When the user signs-in they will have the option of choosing the *AdministratorAccess* role.

Step 6: Okta: Confirm Okta users access to AWS resources

1. Sign in using a test account to the Okta dashboard.
2. Under **My Apps**, select the AWS IAM Identity Center icon.
3. You should see the AWS account icon. Expand that icon to see the list of AWS accounts that the user can access. In this tutorial you only worked with a single account, so expanding the icon only shows one account.
4. Select the account to display the permission sets available to the user. In this tutorial you created the **AdministratorAccess** permission set.
5. Next to the permission set are links for the type of access available for that permission set. When you created the permission set, you specified access to both the AWS Management Console and programmatic access. Select **Management console** to open the AWS Management Console.
6. The user is signed in to the AWS Management Console.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
```

```
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Next steps

Now that you've configured Okta as an identity provider and provisioned users in IAM Identity Center, you can:

- Grant access to AWS accounts, see [Assign user access to AWS accounts](#).
- Grant access to cloud applications, see [Assign user access to applications in the IAM Identity Center console](#).
- Configure permissions based on job functions, see [Create a permission set](#).

Setting up SCIM provisioning between OneLogin and IAM Identity Center

IAM Identity Center supports automatic provisioning (synchronization) of user and group information from OneLogin into IAM Identity Center using the System for Cross-domain Identity Management (SCIM) v2.0 protocol. You configure this connection in OneLogin, using your SCIM endpoint for IAM Identity Center and a bearer token that is created automatically by IAM Identity Center. When you configure SCIM synchronization, you create a mapping of your user attributes in OneLogin to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and OneLogin.

The following steps walk you through how to enable automatic provisioning of users and groups from OneLogin to IAM Identity Center using the SCIM protocol.

Note

Before you begin deploying SCIM, we recommend that you first review the [Considerations for using automatic provisioning](#).

Topics

- [Prerequisites](#)

- [Step 1: Enable provisioning in IAM Identity Center](#)
- [Step 2: Configure provisioning in OneLogin](#)
- [\(Optional\) Step 3: Configure user attributes in OneLogin for access control in IAM Identity Center](#)
- [\(Optional\) Passing attributes for access control](#)
- [Troubleshooting](#)

Prerequisites

You will need the following before you can get started:

- A OneLogin account. If you do not have an existing account, you may be able to obtain a free trial or developer account from the [OneLogin website](#).
- An IAM Identity Center-enabled account ([free](#)). For more information, see [Enable IAM Identity Center](#).
- A SAML connection from your OneLogin account to IAM Identity Center. For more information, see [Enabling Single Sign-On Between OneLogin and AWS](#) on the AWS Partner Network Blog.

Step 1: Enable provisioning in IAM Identity Center

In this first step, you use the IAM Identity Center console to enable automatic provisioning.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

⚠ Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

You have now set up provisioning in the IAM Identity Center console. Now you need to do the remaining tasks using the OneLogin admin console as described in the following procedure.

Step 2: Configure provisioning in OneLogin

Use the following procedure in the OneLogin admin console to enable integration between IAM Identity Center and the IAM Identity Center app. This procedure assumes you have already configured the AWS Single Sign-On application in OneLogin for SAML authentication. If you have not yet created this SAML connection, please do so before proceeding and then return here to complete the SCIM provisioning process. For more information about configuring SAML with OneLogin, see [Enabling Single Sign-On Between OneLogin and AWS](#) on the AWS Partner Network Blog.

To configure provisioning in OneLogin

1. Sign in to OneLogin, and then navigate to **Applications > Applications**.
2. On the **Applications** page, search for the application you created previously to form your SAML connection with IAM Identity Center. Choose it and then choose **Configuration** from the left navigation bar.
3. In the previous procedure, you copied the **SCIM endpoint** value in IAM Identity Center. Paste that value into the **SCIM Base URL** field in OneLogin. Also, in the previous procedure you copied the **Access token** value in IAM Identity Center. Paste that value into the **SCIM Bearer Token** field in OneLogin.
4. Next to **API Connection**, click **Enable**, and then click **Save** to complete the configuration.
5. In the left navigation bar, choose **Provisioning**.
6. Select the check boxes for **Enable provisioning**, **Create user**, **Delete user**, and **Update user**, and then choose **Save**.

7. In the left navigation bar, choose **Users**.
8. Click **More Actions** and choose **Sync logins**. You should receive the message *Synchronizing users with AWS Single Sign-On*.
9. Click **More Actions** again, and then choose **Reapply entitlement mappings**. You should receive the message *Mappings are being reapplied*.
10. At this point, the provisioning process should begin. To confirm this, navigate to **Activity > Events**, and monitor the progress. Successful provisioning events, as well as errors, should appear in the event stream.
11. To verify that your users and groups have all been successfully synchronized to IAM Identity Center, return to the IAM Identity Center console and choose **Users**. Your synchronized users from OneLogin appear on the **Users** page. You can also view your synchronized groups on the **Groups** page.
12. To synchronize user changes automatically to IAM Identity Center, navigate to the **Provisioning** page, locate the **Require admin approval before this action is performed** section, de-select **Create User**, **Delete User**, and/or **Update User**, and click **Save**.

(Optional) Step 3: Configure user attributes in OneLogin for access control in IAM Identity Center

This is an optional procedure for OneLogin if you choose to configure attributes you will use in IAM Identity Center to manage access to your AWS resources. The attributes that you define in OneLogin are passed in a SAML assertion to IAM Identity Center. You will then create a permission set in IAM Identity Center to manage access based on the attributes you passed from OneLogin.

Before you begin this procedure, you must first enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

To configure user attributes in OneLogin for access control in IAM Identity Center

1. Sign in to OneLogin, and then navigate to **Applications > Applications**.
2. On the **Applications** page, search for the application you created previously to form your SAML connection with IAM Identity Center. Choose it and then choose **Parameters** from the left navigation bar.
3. In the **Required Parameters** section, do the following for each attribute you want to use in IAM Identity Center:

- a. Choose **+**.
 - b. In **Field name**, enter `https://aws.amazon.com/SAML/Attributes/AccessControl:AttributeName`, and replace **AttributeName** with the name of the attribute you are expecting in IAM Identity Center. For example, `https://aws.amazon.com/SAML/Attributes/AccessControl:Department`.
 - c. Under **Flags**, check the box next to **Include in SAML assertion**, and choose **Save**.
 - d. In the **Value** field, use the drop-down list to choose the OneLogin user attributes. For example, **Department**.
4. Choose **Save**.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Troubleshooting

The following can help you troubleshoot some common issues you might encounter while setting up automatic provisioning with OneLogin.

Groups are not provisioned to IAM Identity Center

By default, groups may not be provisioned from OneLogin to IAM Identity Center. Ensure that you've enabled group provisioning for your IAM Identity Center application in OneLogin. To do this, sign in to the OneLogin admin console, and check to make sure that the **Include in User Provisioning** option is selected under the properties of the IAM Identity Center application (**IAM Identity Center application > Parameters > Groups**). For more details on how to create groups in OneLogin, including how to synchronize OneLogin roles as groups in SCIM, please see the [OneLogin website](#).

Nothing is synchronized from OneLogin to IAM Identity Center, despite all settings being correct

In addition to the note above regarding admin approval, you will need to **Reapply entitlement mappings** for many configuration changes to take effect. This can be found in **Applications > Applications > IAM Identity Center application > More Actions**. You can see details and logs for most actions in OneLogin, including synchronization events, under **Activity > Events**.

I've deleted or disabled a group in OneLogin, but it still appears in IAM Identity Center

OneLogin currently does not support the SCIM DELETE operation for groups, which means that the group continues to exist in IAM Identity Center. You must therefore remove the group from IAM Identity Center directly to ensure that any corresponding permissions in IAM Identity Center for that group are removed.

I deleted a group in IAM Identity Center without first deleting it from OneLogin and now I'm having user/group sync issues

To remedy this situation, first ensure that you do not have any redundant group provisioning rules or configurations in OneLogin. For example, a group directly assigned to an application along with a rule that publishes to the same group. Next, delete any undesirable groups in IAM Identity Center. Finally, in OneLogin, **Refresh** the entitlements (**IAM Identity Center App > Provisioning > Entitlements**), and then **Reapply entitlement mappings (IAM Identity Center App > More Actions)**. To avoid this issue in the future, first make the change to stop provisioning the group in OneLogin, then delete the group from IAM Identity Center.

Using Ping Identity products with IAM Identity Center

The following Ping Identity products have been tested with IAM Identity Center.

Topics

- [PingFederate](#)
- [PingOne](#)

PingFederate

IAM Identity Center supports automatic provisioning (synchronization) of user and group information from the PingFederate product by Ping Identity (hereafter “Ping”) into IAM Identity Center. This provisioning uses the System for Cross-domain Identity Management (SCIM) v2.0 protocol. You configure this connection in PingFederate using your IAM Identity Center SCIM endpoint and access token. When you configure SCIM synchronization, you create a mapping of your user attributes in PingFederate to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and PingFederate.

This guide is based on PingFederate version 10.2. Steps for other versions may vary. Contact Ping for more information about how to configure provisioning to IAM Identity Center for other versions of PingFederate.

The following steps walk you through how to enable automatic provisioning of users and groups from PingFederate to IAM Identity Center using the SCIM protocol.

Note

Before you begin deploying SCIM, we recommend that you first review the [Considerations for using automatic provisioning](#). Then continue reviewing additional considerations in the next section.

Topics

- [Prerequisites](#)
- [Additional considerations](#)
- [Step 1: Enable provisioning in IAM Identity Center](#)
- [Step 2: Configure provisioning in PingFederate](#)
- [\(Optional\) Step 3: Configure user attributes in PingFederate for access control in IAM Identity Center](#)
- [\(Optional\) Passing attributes for access control](#)

Prerequisites

You will need the following before you can get started:

- A working PingFederate server. If you do not have an existing PingFederate server, you might be able to obtain a free trial or developer account from the [Ping Identity](#) website. The trial includes licenses and software downloads and associated documentation.
- A copy of the PingFederate IAM Identity Center Connector software installed on your PingFederate server. For more information about how to obtain this software, see [IAM Identity Center Connector](#) on the Ping Identity website.
- An IAM Identity Center-enabled account ([free](#)). For more information, see [Enable IAM Identity Center](#).
- A SAML connection from your PingFederate instance to IAM Identity Center. For instructions on how to configure this connection, see the PingFederate documentation. In summary, the recommended path is to use the IAM Identity Center Connector to configure "Browser SSO" in PingFederate, using the "download" and "import" metadata features on both ends to exchange SAML metadata between PingFederate and IAM Identity Center.

Additional considerations

The following are important considerations about PingFederate that can affect how you implement provisioning with IAM Identity Center.

- If an attribute (such as a phone number) is removed from a user in the data store configured in PingFederate, that attribute will not be removed from the corresponding user in IAM Identity Center. This is a known limitation in PingFederate's provisioner implementation. If an attribute is changed to a different (non-empty) value on a user, that change will be synchronized to IAM Identity Center.

Step 1: Enable provisioning in IAM Identity Center

In this first step, you use the IAM Identity Center console to enable automatic provisioning.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.

3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/11111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

 **Warning**

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

Now that you have set up provisioning in the IAM Identity Center console, you must complete the remaining tasks using the PingFederate administrative console. The steps are described in the following procedure.

Step 2: Configure provisioning in PingFederate

Use the following procedure in the PingFederate administrative console to enable integration between IAM Identity Center and the IAM Identity Center Connector. This procedure assumes that you have already installed the IAM Identity Center Connector software. If you have not yet done so, refer to [Prerequisites](#), and then complete this procedure to configure SCIM provisioning.

 **Important**

If your PingFederate server has not been previously configured for outbound SCIM provisioning, you may need to make a configuration file change to enable provisioning. For more information, see Ping documentation. In summary, you must modify the `pf.provisioner.mode` setting in the `pingfederate-<version>/pingfederate/bin/run.properties` file to a value other than OFF (which is the default), and restart the server if

currently running. For example, you may choose to use **STANDALONE** if you don't currently have a high-availability configuration with PingFederate.

To configure provisioning in PingFederate

1. Sign on to the PingFederate administrative console.
2. Select **Applications** from the top of the page, then click **SP Connections**.
3. Locate the application you created previously to form your SAML connection with IAM Identity Center, and click on the connection name.
4. Select **Connection Type** from the dark navigation headings near the top of the page. You should see **Browser SSO** already selected from your previous configuration of SAML. If not, you must complete those steps first before you can continue.
5. Select the **Outbound Provisioning** check box, choose **IAM Identity Center Cloud Connector** as the type, and click **Save**. If **IAM Identity Center Cloud Connector** does not appear as an option, ensure that you have installed the IAM Identity Center Connector and have restarted your PingFederate server.
6. Click **Next** repeatedly until you arrive on the **Outbound Provisioning** page, and then click the **Configure Provisioning** button.
7. In the previous procedure, you copied the **SCIM endpoint** value in IAM Identity Center. Paste that value into the **SCIM URL** field in the PingFederate console. Also, in the previous procedure you copied the **Access token** value in IAM Identity Center. Paste that value into the **Access Token** field in the PingFederate console. Click **Save**.
8. On the **Channel Configuration (Configure Channels)** page, click **Create**.
9. Enter a **Channel Name** for this new provisioning channel (such as **AWSIAMIdentityCenterchannel1**), and click **Next**.
10. On the **Source** page, choose the **Active Data Store** you want to use for your connection to IAM Identity Center, and click **Next**.

Note

If you have not yet configured a data source, you must do so now. See the Ping product documentation for information on how to choose and configure a data source in PingFederate.

11. On the **Source Settings** page, confirm all values are correct for your installation, then click **Next**.
12. On the **Source Location** page, enter settings appropriate to your data source, and then click **Next**. For example, if using Active Directory as an LDAP directory:
 - a. Enter the **Base DN** of your AD forest (such as **DC=myforest,DC=mydomain,DC=com**).
 - b. In **Users > Group DN**, specify a single group that contains all of the users that you want to provision to IAM Identity Center. If no such single group exists, create that group in AD, return to this setting, and then enter the corresponding DN.
 - c. Specify whether to search subgroups (**Nested Search**), and any required LDAP **Filter**.
 - d. In **Groups > Group DN**, specify a single group that contains all of the groups that you want to provision to IAM Identity Center. In many cases, this may be the same DN as you specified in the **Users** section. Enter **Nested Search** and **Filter** values as required.
13. On the **Attribute Mapping** page, ensure the following, and then click **Next**:
 - a. The **userName** field must be mapped to an **Attribute** that is formatted as an email (user@domain.com). It must also match the value that the user will use to log in to Ping. This value in turn is populated in the SAML nameId claim during federated authentication and used for matching to the user in IAM Identity Center. For example, when using Active Directory, you may choose to specify the UserPrincipalName as the **userName**.
 - b. Other fields suffixed with a * must be mapped to attributes that are non-null for your users.
14. On the **Activation & Summary** page, set the **Channel Status** to **Active** to cause the synchronization to start immediately after the configuration is saved.
15. Confirm that all configuration values on the page are correct, and click **Done**.
16. On the **Manage Channels** page, click **Save**.
17. At this point, provisioning starts. To confirm activity, you can view the **provisioner.log** file, located by default in the **pingfederate-<version>/pingfederate/log** directory on your PingFederate server.
18. To verify that users and groups have been successfully synchronized to IAM Identity Center, return to the IAM Identity Center Console and choose **Users**. Synchronized users from PingFederate appear on the **Users** page. You can also view synchronized groups on the **Groups** page.

(Optional) Step 3: Configure user attributes in PingFederate for access control in IAM Identity Center

This is an optional procedure for PingFederate if you choose to configure attributes you will use in IAM Identity Center to manage access to your AWS resources. The attributes that you define in PingFederate are passed in a SAML assertion to IAM Identity Center. You will then create a permission set in IAM Identity Center to manage access based on the attributes you passed from PingFederate.

Before you begin this procedure, you must first enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

To configure user attributes in PingFederate for access control in IAM Identity Center

1. Sign on to the PingFederate administrative console.
2. Choose **Applications** from the top of the page, then click **SP Connections**.
3. Locate the application you created previously to form your SAML connection with IAM Identity Center, and click on the connection name.
4. Choose **Browser SSO** from the dark navigation headings near the top of the page. Then click on **Configure Browser SSO**.
5. On the **Configure Browser SSO** page, choose **Assertion Creation**, and then click on **Configure Assertion Creation**.
6. On the **Configure Assertion Creation** page, choose **Attribute Contract**.
7. On the **Attribute Contract** page, under **Extend the Contract** section, add a new attribute by performing the following steps:
 - a. In the text box, enter `https://aws.amazon.com/SAML/Attributes/AccessControl:AttributeName`, replace **AttributeName** with the name of the attribute you are expecting in IAM Identity Center. For example, `https://aws.amazon.com/SAML/Attributes/AccessControl:Department`.
 - b. For **Attribute Name Format**, choose `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`.
 - c. Choose **Add**, and then choose **Next**.
8. On the **Authentication Source Mapping** page, choose the Adapter Instance configured with your application.

9. On the **Attribute Contract Fulfillment** page, choose the **Source** (*data store*) and **Value** (*data store attribute*) for the **Attribute Contract** `https://aws.amazon.com/SAML/Attributes/AccessControl:Department`.

Note

If you have not yet configured a data source, you will need to do so now. See the Ping product documentation for information on how to choose and configure a data source in PingFederate.

10. Click **Next** repeatedly until you arrive on the **Activation & Summary** page, and then click **Save**.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

PingOne

IAM Identity Center supports automatic provisioning (synchronization) of user information from the PingOne product by Ping Identity (hereafter “Ping”) into IAM Identity Center. This provisioning uses the System for Cross-domain Identity Management (SCIM) v2.0 protocol. You configure this connection in PingOne using your IAM Identity Center SCIM endpoint and access token. When you

configure SCIM synchronization, you create a mapping of your user attributes in PingOne to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and PingOne.

This guide is based on PingOne as of October 2020. Steps for newer versions may vary. Contact Ping for more information about how to configure provisioning to IAM Identity Center for other versions of PingOne. This guide also contains a few notes regarding configuration of user authentication through SAML.

The following steps walk you through how to enable automatic provisioning of users from PingOne to IAM Identity Center using the SCIM protocol.

Note

Before you begin deploying SCIM, we recommend that you first review the [Considerations for using automatic provisioning](#). Then continue reviewing additional considerations in the next section.

Topics

- [Prerequisites](#)
- [Additional considerations](#)
- [Step 1: Enable provisioning in IAM Identity Center](#)
- [Step 2: Configure provisioning in PingOne](#)
- [\(Optional\) Step 3: Configure user attributes in PingOne for access control in IAM Identity Center](#)
- [\(Optional\) Passing attributes for access control](#)

Prerequisites

You will need the following before you can get started:

- A PingOne subscription or free trial, with both federated authentication and provisioning capabilities. For more information about how to obtain a free trial, see the [Ping Identity](#) website.
- An IAM Identity Center-enabled account ([free](#)). For more information, see [Enable IAM Identity Center](#).
- The PingOne IAM Identity Center application added to your PingOne admin portal. You can obtain the PingOne IAM Identity Center application from the PingOne Application Catalog. For

general information, see [Add an application from the Application Catalog](#) on the Ping Identity website.

- A SAML connection from your PingOne instance to IAM Identity Center. After the PingOne IAM Identity Center application has been added to your PingOne admin portal, you must use it to configure a SAML connection from your PingOne instance to IAM Identity Center. Use the “download” and “import” metadata feature on both ends to exchange SAML metadata between PingOne and IAM Identity Center. For instructions on how to configure this connection, see the PingOne documentation.

Additional considerations

The following are important considerations about PingOne that can affect how you implement provisioning with IAM Identity Center.

- As of October 2020, PingOne does not support provisioning of groups through SCIM. Contact Ping for the latest information on group support in SCIM for PingOne.
- Users may continue to be provisioned from PingOne after disabling provisioning in the PingOne admin portal. If you need to terminate provisioning immediately, delete the relevant SCIM bearer token, and/or disable [Provisioning an external identity provider into IAM Identity Center using SCIM](#) in IAM Identity Center.
- If an attribute for a user is removed from the data store configured in PingOne, that attribute will not be removed from the corresponding user in IAM Identity Center. This is a known limitation in PingOne’s provisioner implementation. If an attribute is modified, the change will be synchronized to IAM Identity Center.
- The following are important notes regarding your SAML configuration in PingOne:
 - IAM Identity Center supports only `emailaddress` as a NameId format. This means you need to choose a user attribute that is unique within your directory in PingOne, non-null, and formatted as an email/UPN (for example, `user@domain.com`) for your **SAML_SUBJECT** mapping in PingOne. **Email (Work)** is a reasonable value to use for test configurations with the PingOne built-in directory.
 - Users in PingOne with an email address containing a `+` character may be unable to sign in to IAM Identity Center, with errors such as `'SAML_215'` or `'Invalid input'`. To fix this, in PingOne, choose the **Advanced** option for the **SAML_SUBJECT** mapping in **Attribute Mappings**. Then set **Name ID Format to send to SP:** to `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` in the drop-down menu.

Step 1: Enable provisioning in IAM Identity Center

In this first step, you use the IAM Identity Center console to enable automatic provisioning.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

Now that you have set up provisioning in the IAM Identity Center console, you need to complete the remaining tasks using the PingOne IAM Identity Center application. These steps are described in the following procedure.

Step 2: Configure provisioning in PingOne

Use the following procedure in the PingOne IAM Identity Center application to enable provisioning with IAM Identity Center. This procedure assumes that you have already added the PingOne IAM Identity Center application to your PingOne admin portal. If you have not yet done so, refer to [Prerequisites](#), and then complete this procedure to configure SCIM provisioning.

To configure provisioning in PingOne

1. Open the PingOne IAM Identity Center application that you installed as part of configuring SAML for PingOne (**Applications > My Applications**). See [Prerequisites](#).
2. Scroll to the bottom of the page. Under **User Provisioning**, choose the **complete** link to navigate to the user provisioning configuration of your connection.
3. On the **Provisioning Instructions** page, choose **Continue to Next Step**.
4. In the previous procedure, you copied the **SCIM endpoint** value in IAM Identity Center. Paste that value into the **SCIM URL** field in the PingOne IAM Identity Center application. Also, in the previous procedure you copied the **Access token** value in IAM Identity Center. Paste that value into the **ACCESS_TOKEN** field in the PingOne IAM Identity Center application.
5. For **REMOVE_ACTION**, choose either **Disabled** or **Deleted** (see the description text on the page for more details).
6. On the **Attribute Mapping** page, choose a value to use for the **SAML_SUBJECT** (NameId) assertion, following guidance from [Additional considerations](#) earlier on this page. Then choose **Continue to Next Step**.
7. On the **PingOne App Customization - IAM Identity Center** page, make any desired customization changes (optional), and click **Continue to Next Step**.
8. On the **Group Access** page, choose the groups containing the users you would like to enable for provisioning and single sign-on to IAM Identity Center. Choose **Continue to Next Step**.
9. Scroll to the bottom of the page, and choose **Finish** to start provisioning.
10. To verify that users have been successfully synchronized to IAM Identity Center, return to the IAM Identity Center console and choose **Users**. Synchronized users from PingOne will appear on the **Users** page. These users can now be assigned to accounts and applications within IAM Identity Center.

Remember that PingOne does not support provisioning of groups or group memberships through SCIM. Contact Ping for more information.

(Optional) Step 3: Configure user attributes in PingOne for access control in IAM Identity Center

This is an optional procedure for PingOne if you choose to configure attributes for IAM Identity Center to manage access to your AWS resources. The attributes that you define in PingOne is

passed in a SAML assertion to IAM Identity Center. You then create a permission set in IAM Identity Center to manage access based on the attributes you passed from PingOne.

Before you begin this procedure, you must first enable the [Attributes for access control](#) feature. For more information about how to do this, see [Enable and configure attributes for access control](#).

To configure user attributes in PingOne for access control in IAM Identity Center

1. Open the PingOne IAM Identity Center application that you installed as part of configuring SAML for PingOne (**Applications > My Applications**).
2. Choose **Edit**, and then choose **Continue to Next Step** until you get to the **Attribute Mappings** page.
3. On the **Attribute Mappings** page, choose **Add new attribute**, and then do the following. You must perform these steps for each attribute you will add for use in IAM Identity Center for access control.
 - a. In the **Application Attribute** field, enter `https://aws.amazon.com/SAML/Attributes/AccessControl:AttributeName`. Replace `AttributeName` with the name of the attribute you are expecting in IAM Identity Center. For example, `https://aws.amazon.com/SAML/Attributes/AccessControl:Email`.
 - b. In the **Identity Bridge Attribute or Literal Value** field, choose user attributes from your PingOne directory. For example, **Email (Work)**.
4. Choose **Next** a few times, and then choose **Finish**.

(Optional) Passing attributes for access control

You can optionally use the [Attributes for access control](#) feature in IAM Identity Center to pass an `Attribute` element with the `Name` attribute set to `https://aws.amazon.com/SAML/Attributes/AccessControl:{TagKey}`. This element allows you to pass attributes as session tags in the SAML assertion. For more information about session tags, see [Passing session tags in AWS STS](#) in the *IAM User Guide*.

To pass attributes as session tags, include the `AttributeValue` element that specifies the value of the tag. For example, to pass the tag key-value pair `CostCenter = blue`, use the following attribute.

```
<saml:AttributeStatement>
<saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/AccessControl:CostCenter">
```

```
<saml:AttributeValue>blue
</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
```

If you need to add multiple attributes, include a separate `Attribute` element for each tag.

Configure user access with the default IAM Identity Center directory

When you enable IAM Identity Center for the first time, it's automatically configured with an Identity Center directory as your default identity source, so you don't need to choose an identity source. If your organization uses another identity provider such as AWS Directory Service for Microsoft Active Directory, Microsoft Entra ID, or Okta consider integrating that identity source with IAM Identity Center instead of using the default configuration.

Objective

In this tutorial, you will use the default directory as your identity source and set up and test user access. In this scenario, you manage all users and groups in IAM Identity Center. Users sign in through the AWS access portal. This tutorial is intended for users that are new to AWS or that have been using IAM to manage users and groups. In the next steps, you will create the following:

- An administrative user named *Nikki Wolf*
- A group named *Admin team*
- A permission set named *AdminAccess*

To verify everything was created correctly, you will sign in and set the administrative user's password. After completing this tutorial you can use the administrative user to add more users in IAM Identity Center, create additional permission sets, and set up organizational access to applications.

If you haven't enabled IAM Identity Center yet, see [Enabling AWS IAM Identity Center](#).

Before you begin:

Do either of the following to sign in to the AWS Management Console.

- **New to AWS (root user)** – Sign in as the account owner by choosing **AWS account root user** and entering your AWS account email address. On the next page, enter your password.
- **Already using AWS (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.

Open the [IAM Identity Center console](#).

Step 1: Add a user

1. In the IAM Identity Center navigation pane, choose **Users**, then select **Add user**.
2. On the **Specify user details** page, complete the following information:

- **Username** - For this tutorial, enter *nikkiw*.

When creating users, choose usernames that are easy to remember. Your users must remember the username to sign in to the AWS access portal and you can't change it later.

- **Password** - Choose **Send an email to this user with password setup instructions (Recommended)**.

This option sends the user an email addressed from Amazon Web Services, with the subject line **Invitation to join IAM Identity Center (successor to AWS Single Sign-On)**. The email comes from either `no-reply@signin.aws` or `no-reply@login.awsapps.com`. Add these email addresses to your approved senders list.

- **Email address** - Enter an email address for the user where you can receive the email. Then, enter it again to confirm it. Each user must have a unique email address.
 - **First name** - Enter the first name for the user. For this tutorial, enter *Nikki*.
 - **Last name** - Enter the last name for the user. For this tutorial, enter *Wolf*.
 - **Display name** - The default value is the first and last name of the user. If you want to change the display name, you can enter something different. The display name is visible in the sign-in portal and users list.
 - Complete the optional information if desired. It isn't used during this tutorial and you can change it later.
3. Choose **Next**. The **Add user to groups** page appears. We're going to create a group to assign administrative permissions to instead of giving them directly to *Nikki*.

Choose **Create group**

A new browser tab opens to display the **Create group** page.

- a. Under **Group details**, in **Group name** enter a name for the group. We recommend a group name that identifies the role of the group. For this tutorial, enter *Admin team*.
 - b. Choose **Create group**
 - c. Close the **Groups** browser tab to return to the **Add user** browser tab
4. In the **Groups** area, select the **Refresh** button. The *Admin team* group appears in the list.

Select the checkbox next to *Admin team*, and then choose **Next**.

5. On the **Review and add user** page, confirm the following:
- Primary information appears as you intended
 - Groups shows the user added to the group you created

If you want to make changes, choose **Edit**. When all details are correct choose **Add user**.

A notification message informs you that the user was added.

Next, you will add administrative permissions for the *Admin team* group so that *Nikki* has access to resources.

Step 2: Add administrative permissions

1. In the IAM Identity Center navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
2. On the **AWS accounts** page the **Organizational structure** displays your organization with your accounts underneath it in the hierarchy. Select the checkbox for your management account, then select **Assign users or groups**.
3. The **Assign users and groups** workflow displays. It consists of three steps:
 - a. For **Step 1: Select users and groups** choose the *Admin team* group you created. Then choose **Next**.
 - b. For **Step 2: Select permission sets** choose **Create permission set** to open a new tab that steps you through the three sub-steps involved in creating a permission set.
 - i. For **Step 1: Select permission set type** complete the following:

- In **Permission set type**, choose **Predefined permission set**.
- In **Policy for predefined permission set**, choose **AdministratorAccess**.

Choose **Next**.

- ii. For **Step 2: Specify permission set details**, keep the default settings, and choose **Next**.

The default settings create a permission set named *AdministratorAccess* with session duration set to one hour. You can change the name of the permission set by entering a new name in the **Permission set name** field.

- iii. For **Step 3: Review and create**, verify that the **Permission set type** uses the AWS managed policy **AdministratorAccess**. Choose **Create**. On the **Permission sets** page a notification appears informing you that the permission set was created. You can close this tab in your web browser now.

On the **Assign users and groups** browser tab, you are still on **Step 2: Select permission sets** from which you started the create permission set workflow.

In the **Permissions sets** area, choose the **Refresh** button. The *AdministratorAccess* permission set you created appears in the list. Select the check box for that permission set and then choose **Next**.

- c. On the **Step 3: Review and submit assignments** page, confirm that the *Admin team* group is selected and that the *AdministratorAccess* permission set is selected, then choose **Create**.

The page updates with a message that your AWS account is being configured. Wait until the process completes.

You are returned to the AWS accounts page. A notification message informs you that your AWS account has been reprovisioned and the updated permission set applied.

Congratulations!

You have successfully set up your first user, group, and permission set.

In the next portion of this tutorial you will test *Nikki's* access by signing in to the AWS access portal with their administrative credentials and set their password. Sign out of the console now.

Step 3: Test user access

Now that *Nikki Wolf* is a user in your organization, they can sign in and access the resources to which they're granted permission according to their permission set. To verify that the user is correctly configured, in this next step you will use *Nikki's* credentials to sign in and set up their password. When you added the user *Nikki Wolf* in Step 1 you chose to have *Nikki* receive an email with password setup instructions. It's time to open that email and do the following:

1. In the email, select the **Accept invitation** link to accept the invitation.

Note

The email also includes *Nikki's* user name and the AWS access portal URL that they will use to sign in to the organization. Record this information for future use.

You are taken to the **New user sign up** page where you can set *Nikki's* password.

2. After setting *Nikki's* password, you are navigated to the **Sign in** page. Enter *nikkiw* and choose **Next**, then enter *Nikki's* password and choose **Sign in**.
3. The AWS access portal opens displaying the organization and applications you can access.

Select the organization to expand it into a list of AWS accounts then select the account to display the roles that you can use to access resources in the account.

Each permission set has two management methods you can use, either **Role** or **Access keys**.

- **Role**, for example *AdministratorAccess* - Opens the AWS Console Home.
- **Access keys** - Provides credentials that you can use with the AWS CLI or and AWS SDK. Includes the information for using either short-term credentials that automatically refresh or short-term access keys. For more information, see [Getting IAM Identity Center user credentials for the AWS CLI or AWS SDKs](#).

4. Choose the **Role** link to sign in to the AWS Console Home.

You are signed in and navigated to the AWS Console Home page. Explore the console and confirm that you have the access you expected.

Next steps

Now that you've created an administrative user in IAM Identity Center, you can:

- [Assign applications](#)
- [Add other users](#)
- [Assign users to accounts](#)
- [Configure additional permission sets](#)

Note

You can assign multiple permission sets to the same user. To follow the best practice of applying least-privilege permissions, after you create your administrative user, create a more restrictive permission set and assign it to the same user. That way, you can access your AWS account with only the permissions that you require, rather than administrative permissions.

After your users [accept their invitation](#) to activate their account and they sign into the AWS access portal, the only items that appear in the portal are for the AWS accounts, roles, and applications to which they're assigned.

Important

We strongly recommend that you enable multi-factor authentication (MFA) for your users. For more information, see [Multi-factor authentication for Identity Center users](#).

Get started with common tasks in IAM Identity Center

If you are a new user of IAM Identity Center, the basic workflow to get started using the service is:

1. Sign in to the console of your management account if you are using an organization instance of IAM Identity Center or your AWS account if you are using an account instance of IAM Identity Center and navigate to the IAM Identity Center console.
2. Select your identity source from the IAM Identity Center console. You can connect your existing identity source, such as an [external identity provider](#) or [Active Directory](#). IAM Identity Center also provides you a directory by default that you can use to [configure user access](#).
3. For organization instances, [assign user access to AWS accounts](#) by selecting the accounts in your organization, and then selecting users or groups from your directory and the permissions you want to grant them.
4. Give users access to applications by:
 - a. [Set up customer managed SAML 2.0 applications](#) by either electing one of the pre-integrated applications from the application catalog or adding your own SAML 2.0 application.
 - b. Configure the application properties.
 - c. [Assign the users access](#) to the application. We recommend that you assign user access through group membership rather than by adding individual user permissions. With groups you can grant or deny permissions to groups of users, instead of applying those permissions to each individual. If a user moves to a different organization, you simply move that user to a different group. The user then automatically receives the permissions that are needed for the new organization.
5. If you are using the default IAM Identity Center directory, tell your users how to sign in to the AWS access portal. New users in IAM Identity Center must activate their user credentials before they can be used to sign in to the AWS access portal. For more information, see [Sign in to the AWS access portal](#) in the *AWS Sign-In User Guide*

The topics in this section help familiarize you with the common tasks performed after you have completed the initial configuration of IAM Identity Center.

If you haven't enabled IAM Identity Center yet, see [Enabling AWS IAM Identity Center](#).

Topics

- [Set up single sign-on access to your applications](#)

- [Create a permission set for job functions](#)
- [Assign AWS account access for an IAM Identity Center user](#)
- [Assign AWS account access for groups](#)
- [View user and group assignments](#)
- [Sign in to the AWS access portal with your IAM Identity Center credentials](#)

Set up single sign-on access to your applications

IAM Identity Center supports two application types: *AWS managed applications* and *customer managed applications*.

AWS managed applications are configured directly from within the relevant application consoles or through the application APIs.

Customer managed applications must be added to the IAM Identity Center console and configured with the appropriate metadata for both IAM Identity Center and the service provider. You can choose from a catalog of commonly used applications that support SAML 2.0, or you can set up your own SAML 2.0 applications or OAuth 2.0 applications.

The configuration steps for setting up single sign-on access to applications vary based on the application type.

Set up an AWS managed application

AWS managed applications such as Amazon Managed Grafana and Amazon Monitron integrate with IAM Identity Center. To set up an AWS managed application to work with IAM Identity Center, you must configure the application directly from the console for the applicable service, or you must use the application APIs.


Set up an application from the application catalog

You can select a SAML 2.0 application from a catalog of commonly used applications in the IAM Identity Center console. Use this procedure to set up a SAML 2.0 trust relationship between IAM Identity Center and your application's service provider.

To set up an application from the application catalog

1. Open the [IAM Identity Center console](#).

2. Choose **Applications**.
3. Choose the **Customer managed** tab.
4. Choose **Add application**.
5. On the **Select application type** page, under **Setup preference**, choose **I want to select an application from the catalog**.
6. Under **Application catalog**, start typing the name of the application that you want to add in the search box.
7. Choose the name of the application from the list when it appears in the search results, and then choose **Next**.
8. On the **Configure application** page, the **Display name** and **Description** fields are prepopulated with relevant details for the application. You can edit this information.
9. Under **IAM Identity Center metadata**, do the following:
 - a. Under **IAM Identity Center SAML metadata file**, choose **Download** to download the identity provider metadata.
 - b. Under **IAM Identity Center certificate**, choose **Download certificate** to download the identity provider certificate.

 **Note**


You will need these files later when you set up the application from the service provider's website. Follow the instructions from that provider.

10. (Optional) Under **Application properties**, you can specify the **Application start URL**, **Relay state**, and **Session duration**. For more information, see [Application properties in the IAM Identity Center console](#).
11. Under **Application metadata**, do one of the following:
 - a. If you have a metadata file, choose **Upload application SAML metadata file**. Then, select **Choose file** to find and select the metadata file.
 - b. If you don't have a metadata file, choose **Manually type your metadata values**, and then provide the **Application ACS URL** and **Application SAML audience** values.
12. Choose **Submit**. You're taken to the details page of the application that you just added.

Set up your own SAML 2.0 application

Use this procedure to set up your own SAML 2.0 trust relationship between IAM Identity Center and your own SAML 2.0 application's service provider. Before you begin this procedure, make sure that you have the service provider's certificate and metadata exchange files so that you can finish setting up the trust.

To set up your own SAML 2.0 application

1. Open the [IAM Identity Center console](#).
 2. Choose **Applications**.
 3. Choose the **Customer managed** tab.
 4. Choose **Add application**.
 5. On the **Select application type** page, under **Setup preference**, choose **I have an application I want to set up**.
 6. Under **Application type**, choose **SAML 2.0**.
 7. Choose **Next**.
 8. On the **Configure application** page, under **Configure application**, enter a **Display name** for the application, such as **MyApp**. Then, enter a **Description**.
 9. Under **IAM Identity Center metadata**, do the following:
 - a. Under **IAM Identity Center SAML metadata file**, choose **Download** to download the identity provider metadata.
 - b. Under **IAM Identity Center certificate**, choose **Download** to download the identity provider certificate.
- 
- Note**
- You will need these files later when you set up the custom application from the service provider's website.
10. (Optional) Under **Application properties**, you can also specify the **Application start URL**, **Relay state**, and **Session duration**. For more information, see [Application properties in the IAM Identity Center console](#).
 11. Under **Application metadata**, choose **Manually type your metadata values**. Then, provide the **Application ACS URL** and **Application SAML audience** values.

12. Choose **Submit**. You're taken to the details page of the application that you just added.

After you have set up your applications, your users can access your applications from within their AWS access portal based on the permissions that you assigned.

If you have customer managed applications that support OAuth 2.0 and your users need access from these applications to AWS services, you can use trusted identity propagation. With trusted identity propagation, a user can sign in to an application, and that application can pass the users' identity in requests to access data in AWS services. For more information, see [Using trusted identity propagation with customer managed applications](#).

For more information about supported application types, see [Manage access to applications](#).

Create a permission set for job functions

Permission sets are stored in IAM Identity Center and define the level of access that users and groups have to an AWS account. The first permission set you create is the administrative permission set. If you completed one of the [Getting started tutorials for IAM Identity Center](#) you already created your administrative permission set. Use this procedure to create permission sets as described in the [AWS managed policies for job functions](#) topic in the *IAM User Guide*.

1. Do either of the following to sign in to the AWS Management Console.
 - **New to AWS (root user)** – Sign in as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
 - **Already using AWS (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. Open the [IAM Identity Center console](#).
3. In the IAM Identity Center navigation pane, under **Multi-account permissions**, choose **Permission sets**.
4. Choose **Create permission set**.
 - a. On the **Select permission set type** page, in the **Permission set type** section, choose **Predefined permission set**.
 - b. In the **Policy for predefined permission set** section, choose one of the following:
 - **AdministratorAccess**

- **Billing**
 - **DatabaseAdministrator**
 - **DataScientist**
 - **NetworkAdministrator**
 - **PowerUserAccess**
 - **ReadOnlyAccess**
 - **SecurityAudit**
 - **SupportUser**
 - **SystemAdministrator**
 - **ViewOnlyAccess**
5. On the **Specify permission set details** page, keep the default settings and choose **Next**. The default setting limits your session to one hour.
 6. On the **Review and create** page, confirm the following:
 1. For **Step 1: Select permission set type**, displays the type of permission set you chose.
 2. For **Step 2: Define permission set details**, displays the name of the permission set you chose.
 3. Choose **Create**.

Create a permission set that applies least-privilege permissions

To follow the best practice of applying least-privilege permissions, after you create an administrative permission set, you create a more restrictive permission set and assign it to one or more users. The permission sets created in the previous procedure provide a starting point for you to assess the amount of access to resources your users need. To switch to least privilege permissions, you can run IAM Access Analyzer to monitor principals with AWS managed policies. After learning which permissions they're using, then you can write a custom policy or generate a policy with only the required permissions for your team.

With IAM Identity Center, you can assign multiple permission sets to the same user. Your administrative user should also be assigned additional, more restrictive, permission sets. That way, they can access your AWS account with only the permissions that required, rather than always using their administrative permissions.

For example, if you're a developer, after you create your administrative user in IAM Identity Center, you can create a new permission set that grants `PowerUserAccess` permissions, and then assign that permission set to yourself. Unlike the administrative permission set, which uses `AdministratorAccess` permissions, the `PowerUserAccess` permission set doesn't allow management of IAM users and groups. When you sign into the AWS access portal to access your AWS account, you can choose `PowerUserAccess` rather than the `AdministratorAccess` to perform development tasks in the account.

Keep the following considerations in mind:

- **To get started quickly with creating a more restrictive permission set, use a predefined permission set rather than a custom permission set.**

With a predefined permission set, which uses [predefined permissions](#), you choose a single AWS managed policy from a list of available policies. Each policy grants a specific level of access to AWS services and resources or permissions for a common job function. For information about each of these policies, see [AWS managed policies for job functions](#).

- **You can configure the session duration for a permission set to control the length of time that a user is signed into an AWS account.**

When users federate into their AWS account and use the AWS Management Console or the AWS Command Line Interface (AWS CLI), IAM Identity Center uses the session duration setting on the permission set to control the duration of the session. By default, the value for **Session duration**, which determines the length of time that a user can be signed into an AWS account before AWS signs the user out of the session, is set to one hour. You can specify a maximum value of 12 hours. For more information, see [Set session duration for AWS accounts](#).

- **You can also configure the AWS access portal session duration to control the length of time that a workforce user is signed into the portal.**

By default, the value for **Maximum session duration**, which determines the length of time that a workforce user can be signed in to the AWS access portal before they must re-authenticate, is eight hours. You can specify a maximum value of 90 days. For more information, see [Configure the session duration of the AWS access portal and IAM Identity Center integrated applications](#).

- **When you sign into the AWS access portal, choose the role that provides least-privilege permissions.**

Each permission set that you create and assign to your user appears as an available role in the AWS access portal. When you sign in to the portal as that user, choose the role that corresponds

to the most restrictive permission set that you can use to perform tasks in the account, rather than `AdministratorAccess`.

- **You can add other users to IAM Identity Center and assign existing or new permission sets to those users.**

For information, see, [Assign AWS account access for groups](#).

Assign AWS account access for an IAM Identity Center user

To set up AWS account access for an IAM Identity Center user, you must assign the user to the AWS account and permission set.

1. Do either of the following to sign in to the AWS Management Console.
 - **New to AWS (root user)** – Sign in as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
 - **Already using AWS (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. Open the [IAM Identity Center console](#).
3. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
4. On the **AWS accounts** page, a tree view list of your organization displays. Select the checkbox next to the AWS account to which you want to assign access. If you are setting up administrative access for IAM Identity Center, select the checkbox next to the management account .
5. Choose **Assign users or groups**.
6. For **Step 1: Select users and groups**, on the **Assign users and groups to "AWS account name"** page, do the following:
 1. On the **Users** tab, select the user to whom you want to grant administrative permissions.

To filter the results, start typing the name of the user that you want in the search box.
 2. After you confirm that the correct user is selected, choose **Next**.
7. For **Step 2: Select permission sets**, on the **Assign permission sets to "AWS account name"** page, under **Permission sets**, select a permission set to define the level of access that users and groups have to this AWS account.
8. Choose **Next**.

9. For **Step 3: Review and Submit**, on the **Review and submit assignments to "AWS account name"** page, do the following:


1. Review the selected user and permission set.
2. After you confirm that the correct user is assigned to the permission set, choose **Submit**.

 **Important**

The user assignment process might take a few minutes to complete. Leave this page open until the process successfully completes.

10. If either of the following applies, follow the steps in [Prompt users for MFA](#) to enable MFA for IAM Identity Center:

- You're using the default Identity Center directory as your identity source.
- You're using an AWS Managed Microsoft AD directory or a self-managed directory in Active Directory as your identity source and you're not using RADIUS MFA with AWS Directory Service.

 **Note**

If you're using an external identity provider, note that the external IdP, not IAM Identity Center, manages MFA settings. MFA in IAM Identity Center is not supported for use by external IdPs.

When you set up account access for the administrative user, IAM Identity Center creates a corresponding IAM role. This role, which is controlled by IAM Identity Center, is created in the relevant AWS account, and the policies specified in the permission set are attached to the role.

Assign AWS account access for groups

After you've created an administrative user in IAM Identity Center and created additional permission sets that you can use to perform tasks with least-privileged permissions, you can provide access to your AWS accounts to user groups.

We recommend that you assign access directly to groups rather than to individual users. For example, if you create groups and permission sets based on organizational units, if a user moves to a different organizational unit, you simply move that user to a different group and they automatically receive the permissions that are needed for the new organizational unit and lose the permissions of the previous organizational unit.

To assign user group access to AWS accounts

1. Open the [IAM Identity Center console](#).

Note

If your identity source is AWS Managed Microsoft AD make sure that the IAM Identity Center console is using the Region where your AWS Managed Microsoft AD directory is located before you move to the next step.

2. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
3. On the **AWS accounts** page, a tree view list of your organization appears. Select the checkbox next to one or more AWS accounts to which you want to assign single sign-on access.

Note

You can select up to 10 AWS accounts per permission set.

4. Choose **Assign users or groups**.
5. For **Step 1: Select users and groups**, on the **Assign users and groups to "AWS-account-name"** page, select the **Groups** tab, then choose one or more groups.

To filter the results, start typing the name of the group that you want in the search box.

To display the groups that you selected, choose the sideways triangle next to **Selected users and groups**.

After you confirm that the correct groups are selected, choose **Next**.

6. For **Step 2: Select permission sets**, on the **Assign permission sets to "AWS-account-name"** page, select one or more permission sets

Note

If you didn't create the permission set you want before starting this procedure choose **Create permission set**, and follow the steps in [Create a permission set](#). After you create the permission sets that you want to apply, in the IAM Identity Center console, return to **AWS accounts** and follow the instructions until you reach **Step 2: Select permission sets**. When you reach this step, select the new permission sets that you created, and proceed to the next step in this procedure.

After you confirm that the correct permission sets are selected, choose **Next**.

7. For **Step 3: Review and Submit**, on the **Review and submit assignments to "AWS-account-name"** page, do the following:
 1. Review the selected groups, and permission sets.
 2. After you confirm that the correct groups, and permission sets are selected, choose **Submit**.

Important

The group assignment process might take a few minutes to complete. Leave this page open until the process successfully completes.

Note

You might need to grant users or groups permissions to operate in the AWS Organizations management account. Because it is a highly privileged account, additional security restrictions require you to have the [IAMFullAccess](#) policy or equivalent permissions before you can set this up. These additional security restrictions are not required for any of the member accounts in your AWS organization.

Alternatively, you can use [AWS CloudFormation](#) to create and assign permission sets and assign users to those permission sets. Users can then [sign in to the AWS access portal](#) or use [AWS Command Line Interface \(AWS CLI\)](#) commands.

View user and group assignments

You can see who has access to what in IAM Identity Center from the **Users** and **Groups** pages. Use this procedure to view the level of access that users have to AWS accounts, permission sets, applications, and groups.

1. Open the [IAM Identity Center console](#).
2. Choose **Users** or **Groups** based on whether you want to edit a group of users or one user that was assigned individually.
3. Choose a user or group from the list.
4. Choose whether you want to view account assignments, application assignments, or group assignments:
 - **AWS account and permission set assignments**
 1. Choose the **Accounts** tab.
 2. Select an account from the list to view user and group permission set assignments.
 3. Select a permission set to view to view policy and assignment details.
 - **Application assignments**
 1. Choose the **Applications** tab to view which applications are assigned to a user or group.
 2. Select an application from the list to view assignment details.
 - **Group assignments**
 1. From the Users page, choose the **Groups** tab.
 2. Select a group to view group assignments for a user.

Sign in to the AWS access portal with your IAM Identity Center credentials

The AWS access portal provides IAM Identity Center users with single sign-on access to all their assigned AWS accounts and applications through a web portal.

Complete the following steps to confirm that the IAM Identity Center user can sign in to the AWS access portal and access the AWS account.

1. Do either of the following to sign in to the AWS Management Console.

- **New to AWS (root user)** – Sign in as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
 - **Already using AWS (IAM credentials)** – Sign in with your IAM credentials and select an admin role.
2. Open the [IAM Identity Center console](#).
 3. In the navigation pane, choose **Dashboard**.
 4. On the **Dashboard** page, under **Settings summary**, choose the AWS access portal URL.
 5. Sign in by using either of the following:
 - If you're using Active Directory or an external identity provider (IdP) as your identity source, sign in by using the credentials of the Active Directory or IdP user.
 - If you're using the default Identity Center directory as your identity source, sign in by using the username that you specified when you created the user and the new password that you specified for the user.
 1. In the **Accounts** tab, locate your AWS account and expand it.
 2. The roles available to you are displayed. For example, if you're assigned both the **AdministratorAccess** permission set and **Billing** permissions sets, those roles are displayed in the AWS access portal. Choose the IAM role name you want to use for the session.
 3. If you're redirected to the AWS Management Console you successfully finished setting up access to the AWS account.

Note

If you don't see any **AWS accounts** listed, it's likely that the user hasn't yet been assigned to a permission set for that account. For instructions on assigning users to a permission set, see [Assign user access to AWS accounts](#).

Now that you've confirmed that you can sign in using IAM Identity Center credentials, switch to the browser that you used to sign into the AWS Management Console and sign out from your root user or IAM user credentials.

 Important

We strongly recommend that you use the credentials of the IAM Identity Center administrative user when you sign in to the AWS access portal to perform administrative tasks instead of using IAM user or root user credentials. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. To enable other users to access your accounts and applications, and to administer IAM Identity Center, create and assign permission sets only through IAM Identity Center.







Manage organization and account instances of IAM Identity Center




An instance is a single deployment of IAM Identity Center. There are two types of instances available for IAM Identity Center: *organization instances* and *account instances*.

AWS account types that can enable IAM Identity Center

To enable IAM Identity Center, sign in to the AWS Management Console by using one of the following credentials, depending on the instance type you want to create:

- **Your AWS Organizations management account (recommended)** – Required to create an [organization instance](#) of IAM Identity Center. Use an organization instance for multi-account permissions and application assignments across the organization.
- **Your AWS Organizations member account** – Use to create an [account instance](#) of IAM Identity Center to enable application assignments within that member account. One or more accounts with a member level instance can exist in an organization.
- **A standalone AWS account** – Use to create an [organization instance](#) or [account instance](#) of IAM Identity Center. The standalone AWS account isn't managed by AWS Organizations. Only one instance of IAM Identity Center can be associated with a standalone AWS account and you can use the instance for application assignments within that standalone AWS account.

Capability	Instance in the AWS Organizations management account (recommended)	Instance in a member account	Instance in a standalone AWS account
Manage users			
AWS access portal for single-sign on access to your AWS			

Capability	Instance in the AWS Organizations management account (recommended)	Instance in a member account	Instance in a standalone AWS account
managed applications			
OAuth 2.0 (OIDC) customer managed applications	 Yes	 Yes	 Yes
Multi-account permissions	 Yes	 No	 No
AWS access portal for single-sign on access to your AWS accounts	 Yes	 No	 No
SAML 2.0 customer managed applications	 Yes	 No	 No
Delegated administrator can manage instance	 Yes	 No	 No

Topics

- [Organization instances of IAM Identity Center](#)
- [Account instances of IAM Identity Center](#)
- [Enable account instance creation in the IAM Identity Center console](#)
- [Control account instance creation with Services Control Policies](#)
- [Create an account instance of IAM Identity Center](#)

- [Delete your IAM Identity Center instance](#)

Organization instances of IAM Identity Center

When you enable IAM Identity Center in conjunction with AWS Organizations, you're creating an organization instance of IAM Identity Center. Your organization instance must be enabled in your management account and you can centrally manage the access of users and groups with a single organization instance. You can have only one organization instance for each management account in AWS Organizations.

If you enabled IAM Identity Center prior to November 15, 2023, you have an organization instance of IAM Identity Center.

To enable an organization instance of IAM Identity Center, see [Enabling AWS IAM Identity Center](#).

When to use an organization instance

An organization instance is the primary method of enabling IAM Identity Center and in most cases, an organization instance is recommended. Organization instances offer the following benefits:

- **Support for all features of IAM Identity Center** – Including managing permissions for multiple AWS accounts in your organization and assigning access to customer managed applications.
- **Reduce the number of management points** – An organization instance has a single management point, the management account. We recommend that you enable an organization instance, rather than an account instance, to reduce the number of management points.
- **Control creation of account instances** – You can control whether account instances can be created by member accounts in your organization as long as you haven't deployed an instance of IAM Identity Center to your organization in an opt-in Region (AWS Region that is disabled by default).

Account instances of IAM Identity Center

With an account instance of IAM Identity Center, you can deploy supported AWS managed applications and OIDC-based customer managed applications. Account instances support isolated deployments of applications in a single AWS account, leveraging IAM Identity Center workforce identity and access portal features.

Account instances are bound to a single AWS account and are used only to manage user and group access for supported applications in the same account and AWS Region. You are limited to one account instance per AWS account. You can create an account instance from either of the following:

- A member account in AWS Organizations.
- A standalone AWS account that is not managed by AWS Organizations.

Topics

- [Availability constraints for member accounts](#)
- [When to use account instances](#)
- [Account instance considerations](#)
- [AWS managed applications that support account instances](#)

Availability constraints for member accounts

You can deploy account instances of IAM Identity Center in AWS Organizations member accounts whether or not an organization instance of IAM Identity Center is already present in the AWS organization.

One of the following conditions must be true:

- There is no organization instance of IAM Identity Center in your AWS organization.
- There is an organization instance of IAM Identity Center in your AWS organization and the instance administrator has enabled member accounts to create account instances of IAM Identity Center (for organization instances created after November 15, 2023).
- There is an organization instance of IAM Identity Center in your AWS organization and the instance administrator manually enabled creation of account instances by member accounts in the organization (for organization instances created before November 15, 2023). For instructions, see [Enable account instance creation in the IAM Identity Center console](#).

After one of the preceding conditions is met, all of the following conditions must be true:

- Your administrator hasn't created a [Service Control Policy](#) that prevents member accounts from creating account instances.

- You don't already have an instance of IAM Identity Center in this same account regardless of AWS Region.
- You're working in an AWS Region where IAM Identity Center is available. For information about Regions, see [AWS IAM Identity Center Region availability](#).

When to use account instances

In most cases, an [organization instance](#) is recommended. Account instances should be used only if one of the following scenarios applies:

- You want to run a temporary trial of a supported AWS managed application to determine if the application suits your business needs.
- You don't have plans to adopt IAM Identity Center across your organization, but you want to support one or more AWS managed applications.
- You have an organization instance of IAM Identity Center, but you want to deploy a supported AWS managed application to an isolated set of users that are distinct from users in your organization instance.

Important

If you plan to use IAM Identity Center to support applications in multiple accounts, create an organization instance and don't use account instances.

Account instance considerations

An account instance is designed for specialized use cases, offering a subset of features available to an organization instance. Consider the following before creating an account instance:

- Account instances don't support permission sets and therefore don't support access to AWS accounts.
- You can't convert an account instance into an organization instance.
- You can't merge an account instance into an organization instance.
- Only select [AWS managed applications](#) support account instances.

- Use account instances for isolated users that will use applications in a single account only and for the lifetime of the applications used.
- Applications that are attached to an account instance must remain attached to the account instance until you delete the application and its resources.
- An account instance must remain in the AWS account where it's created.

AWS managed applications that support account instances

See [AWS managed applications](#) to learn which AWS managed applications support account instances of IAM Identity Center. Verify the availability of account instance creation with your AWS managed application.

Enable account instance creation in the IAM Identity Center console

If you enabled IAM Identity Center before November 15, 2023, you have an [organization instance](#) of IAM Identity Center and the ability for member accounts to create account instances is disabled by default. You can choose whether your member accounts can create account instances by enabling the account instance feature in the IAM Identity Center console.

Note

Member accounts can create an account instance as long as you haven't deployed an instance of IAM Identity Center to your organization in an opt-in Region (AWS Region that's disabled by default) regardless of deployment date. Any organization instance of IAM Identity Center deployed in an opt-in AWS Region will prevent the creation of account instances. For information about opt-in Regions, see [AWS IAM Identity Center Region availability](#).

To enable creation of account instances by member accounts in your organization

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**, and then choose the **Management** tab.
3. In the **Account instances of IAM Identity Center** section, choose **Enable account instances of IAM Identity Center**.

4. In the **Enable account instances of IAM Identity Center** dialog box, confirm that you want to allow member accounts in your organization to create account instances by choosing **Enable**.

⚠ Important

Enabling account instances of IAM Identity Center for member accounts is a one-time operation. This means that this operation can't be reversed. Once enabled, you can limit the creation of account instances by creating a service control policy (SCP). For instructions, see [Control account instance creation with Services Control Policies](#).

Control account instance creation with Services Control Policies

Users can create an instance of IAM Identity Center that's bound to a single AWS account, called an [account instance of IAM Identity Center](#). You can control account instance creation with Service Control Policies (SCP).

1. Open the [IAM Identity Center console](#).
2. On the **Dashboard**, in the **Central management** section, choose the **Prevent account instances** button.
3. In the **Attach SCP to prevent creation of new account instances** dialog box, an SCP is provided for you. Copy the SCP and choose the **Go to SCP dashboard** button. You'll be directed to the [AWS Organizations console](#) to create the SCP or attach it as a statement to an existing SCP.

Service control policies are a feature of AWS Organizations. For instructions on attaching an SCP, see [Attaching and detaching service control policies](#) in the *AWS Organizations User Guide*.

Rather than prevent account instance creation, you can limit account instance creation to a specific AWS account within your organization:

Example : SCP to control instance creation

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

```
    "Effect": "Deny",
    "Action": "sso:CreateInstance",
    "Resource": "*",
    "Condition": {
      "StringNotEquals": {
        "aws:PrincipalAccount": ["<ALLOWED-ACCOUNT-ID>"]
      }
    }
  ]
}
```

Create an account instance of IAM Identity Center

An [organization instance](#) is the primary and recommended method of enabling IAM Identity Center. Make sure your use case supports creating an [account instance](#) and that you're aware of the considerations.

Create an account instance from an organization member account or standalone AWS account

1. Do either of the following to sign in to the AWS Management Console.
 - **New to AWS (root user)** – Sign in as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
 - **Already using AWS (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. Open the [IAM Identity Center console](#).
3. Under **Enable IAM Identity Center**, choose **Enable**.
4. Select **Continue creating the account instance** and choose **Continue**.

Note

If an organization instance of IAM Identity Center exists, ensure that your use case requires its own account instance of IAM Identity Center. If it doesn't, choose **Cancel and use organization instance**.

5. **Optional.** Add tags that you want to associate with this account instance.

A notification in the console indicates a successful account instance is created and includes the instance ID. You can name your instance in the **Settings summary**.





Note




Multi-factor authentication (MFA) is enabled by default for account instances. Users are prompted to sign in with MFA when their device, browser, or location changes. As a security best practice, we strongly recommend MFA for your workforce identities. Learn how to [Manage MFA devices in IAM Identity Center](#).

Management features such as [confirming your identity source](#), [adjusting multi-factor authentication settings](#), and [adding AWS managed applications](#) must be completed in the IAM Identity Center console.

Delete your IAM Identity Center instance

When an IAM Identity Center instance is deleted, all the data in that instance is deleted and can't be recovered. The following table describes what data is deleted based on the directory type that's configured in IAM Identity Center.

What data gets deleted	Connected directory - AWS Managed Microsoft AD, AD Connector, or external identity provider	IAM Identity Center identity store	
All permission sets you have configured for AWS accounts			Yes
All applications you have configured in IAM Identity Center			Yes

What data gets deleted	Connected directory - AWS Managed Microsoft AD, AD Connector, or external identity provider	IAM Identity Center identity store	
All user assignments you have configured for AWS accounts and applications			Yes
All users and groups in the directory or store	N/A		Yes

Use the following procedure to delete your IAM Identity Center instance.

To delete your IAM Identity Center instance

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Management** tab.
4. In the **Delete IAM Identity Center configuration** section, choose **Delete**.
5. In the **Delete IAM Identity Center configuration** dialog, select each of the check boxes to acknowledge you understand that your data that will be deleted. Type your IAM Identity Center instance in the text box, and then choose **Confirm**.

Authentication in IAM Identity Center

A user signs in to the AWS access portal using their user name. When they do, IAM Identity Center redirects the request to the IAM Identity Center authentication service based on the directory associated with the user email address. Once authenticated, users have single sign-on access to any of the AWS accounts and third-party software-as-a-service (SaaS) applications that show up in the portal without additional sign-in prompts. This means that users no longer need to keep track of multiple account credentials for the various assigned AWS applications that they use on a daily basis.

Authentication sessions

There are two types of authentication sessions maintained by IAM Identity Center: one to represent the users' sign in to IAM Identity Center, and another to represent the users' access to AWS managed applications, such as Amazon SageMaker Studio or Amazon Managed Grafana. Each time a user signs in to IAM Identity Center, a sign in session is created for the duration configured in IAM Identity Center, which can be up to 90 days. For more information, see [Configure the session duration of the AWS access portal and IAM Identity Center integrated applications](#). Each time the user accesses an application, the IAM Identity Center sign in session is used to obtain an IAM Identity Center application session for that application. IAM Identity Center application sessions have a refreshable 1-hour lifetime – that is, IAM Identity Center application sessions are automatically refreshed every hour as long as the IAM Identity Center sign in session from which they were obtained is still valid. If the user signs out using the AWS access portal, the user's sign in session ends. The next time application refreshes its session, the application session will end.

When the user uses IAM Identity Center to access the AWS Management Console or CLI, the IAM Identity Center sign in session is used to obtain an IAM session, as specified in the corresponding IAM Identity Center permission set (more specifically, IAM Identity Center assumes an IAM role, which IAM Identity Center manages, in the target account). IAM sessions persist for the time specified for the permission set, unconditionally.

Note

IAM Identity Center does not support SAML Single Logout initiated by an identity provider that acts as your identity source, and it does not send SAML Single Logout to SAML applications that use IAM Identity Center as an Identity provider.

When you disable or delete a user in IAM Identity Center, that user will immediately be prevented from signing in to create new IAM Identity Center sign in sessions. IAM Identity Center sign in sessions are cached for one hour, which means that when you disable or delete a user while they have an active IAM Identity Center sign in session, their existing IAM Identity Center sign in session will continue for up to an hour, depending on when the sign in session was last refreshed. During this time, the user can initiate new IAM Identity Center application and IAM role sessions.

After the IAM Identity Center sign in session expires, the user can no longer initiate new IAM Identity Center application or IAM role sessions. However, IAM Identity Center application sessions can also be cached for up to an hour, such that the user might retain access to an application for up to an hour after the IAM Identity Center sign in session has expired. Any existing IAM role sessions will continue based on the duration configured in the IAM Identity Center permission set (admin-configurable, up to 12 hours).

The table below summarizes these behaviors:

User experience / system behavior	Time after user is disabled / deleted
User can no longer sign in to IAM Identity Center; user cannot obtain a new IAM Identity Center sign in session	None (effective immediately)
User can no longer start new application or IAM role sessions via IAM Identity Center	Up to 1 hour
User can no longer access any applications (all application sessions are terminated)	Up to 2 hours (up to 1 hour for IAM Identity Center sign in session expiration, plus up to 1 hour for IAM Identity Center application session expiration)
User can no longer access any AWS accounts through IAM Identity Center	Up to 13 hours (up to 1 hour for IAM Identity Center sign in session expiry, plus up to 12 hours for administrator-configured IAM role session expiry per the IAM Identity Center session duration settings for the permission set)

For more information about sessions, see [Set session duration for AWS accounts](#).

Connect workforce users

IAM Identity Center is the AWS solution for connecting your workforce users to AWS managed applications such as Amazon Q Developer and Amazon QuickSight, and other AWS resources. You can connect your existing identity provider and synchronize users and groups from your directory, or create and manage your users directly in IAM Identity Center.

Already using IAM for access to AWS accounts?

You don't need to make any changes to your current AWS account workflows to use IAM Identity Center for access to AWS managed applications. If you're using [federation with IAM](#) or IAM users for AWS account access, your users can continue to access AWS accounts in the same way they always have, and you can continue to use your existing workflows to manage that access.

Topics

- [IAM Identity Center use cases](#)
- [Users, groups, and provisioning in IAM Identity Center](#)
- [Manage your identity source](#)
- [Using the AWS access portal](#)
- [Multi-factor authentication for Identity Center users](#)

IAM Identity Center use cases

Learn how to use IAM Identity Center to centrally manage user access and permissions across multiple AWS accounts and various AWS managed and customer managed applications

Use case	Learn more
Use IAM Identity Center to grant application access and administrative permissions to users and groups within an organization.	Enable single sign-on access to your AWS applications (Application admin role)
Use IAM Identity Center to provide users with access to Amazon EC2 instances and manage their permissions.	Enable single sign-on access to your Amazon EC2 Windows instances

Use case	Learn more
Use IAM Identity Center to centrally manage access and permissions for AWS managed and customer managed applications.	Manage access to applications
Use IAM Identity Center to centrally manage access and permissions across multiple AWS accounts within an organization.	Manage access to AWS accounts

Enable single sign-on access to your AWS applications (Application admin role)

This use case provides guidance if you're an application administrator who manages [AWS managed applications](#) such as Amazon SageMaker or AWS IoT SiteWise, and you must provide single sign-on access to your users.

Before you get started, consider the following:

- Do you want to create a test or production environment in a separate organization in AWS Organizations?
- Is IAM Identity Center already enabled in your organization? Do you have permissions to enable IAM Identity Center in the management account of AWS Organizations?

Review the following guidance to determine next steps based on your business needs.

Configure my AWS application in a standalone AWS account

If you must provide single sign-on access to an AWS application and know that your IT department does not yet use IAM Identity Center, you might need to create a standalone AWS account to get started. By default, when you create your own AWS account, you'll have the permissions that you require to create and manage your own AWS organization. To enable IAM Identity Center, you must have AWS account root user permissions.

IAM Identity Center and AWS Organizations can be enabled automatically during setup for some AWS applications (for example, Amazon Managed Grafana). If your AWS application doesn't

provide the option to enable these services, you must set up AWS Organizations and IAM Identity Center before you can provide single sign-on access to your application.

IAM Identity Center isn't configured in my organization

In your role as an application administrator, you might not be able to enable IAM Identity Center, depending on your permissions. IAM Identity Center requires specific permissions in the AWS Organizations management account. In this case, contact the appropriate administrator to have IAM Identity Center enabled in the Organizations management account.

If you do have sufficient permissions to enable IAM Identity Center, do this first, then proceed with the application setup. For more information, see [Get started with common tasks in IAM Identity Center](#).

IAM Identity Center is currently configured in my organization

In this scenario, you can continue to deploy your AWS application without taking any further action.

Note

If your organization enabled IAM Identity Center in the management account before November 25th, 2019, you must also enable AWS managed applications in the management account and optionally in the member accounts. If you enable them in the management account only, you can enable them in member accounts later. To enable these applications, choose **Enable access** in the IAM Identity Center console's **Settings** page in the AWS managed applications section. For more information, see [Configuring IAM Identity Center to share identity information](#).

Enable single sign-on access to your Amazon EC2 Windows instances

You can enable single sign-on access to your Amazon EC2 Windows instances if you're an application administrator who manages users in the Identity Center directory (the default identity source for IAM Identity Center) or a supported external identity provider (IdP), and you must provide IAM Identity Center access to your Amazon EC2 Windows desktops from the AWS Fleet Manager console.

With this configuration, you can securely access your Amazon EC2 Windows instances with existing corporate credentials. You don't need to share administrator credentials, access credentials

multiple times, or configure remote access client software. You can centrally grant and revoke access to your Amazon EC2 Windows instances at scale across multiple AWS accounts. For example, if you remove an employee from your IAM Identity Center integrated identity source, they automatically lose access to all AWS resources, including Amazon EC2 Windows instances.

For more information, see [How to enable secure seamless single sign-on to Amazon EC2 Windows instances with IAM Identity Center](#).

For a demonstration of how to configure IAM Identity Center to enable this capability, see [Enabling Single Sign-on to Amazon EC2 Windows with IAM Identity Center](#).

Users, groups, and provisioning in IAM Identity Center

Keep the following considerations in mind when you work with users and groups in IAM Identity Center.

User name and email address uniqueness

Users in IAM Identity Center must be uniquely identifiable. IAM Identity Center implements a user name that is the primary identifier for your users. Although most people set the user name equal to a user's email address, IAM Identity Center and the SAML 2.0 standard do not require this. However, many SAML 2.0-based applications use an email address as the unique identifier for users. These applications obtain this information from assertions that a SAML 2.0 identity provider sends during authentication. Such applications depend on the uniqueness of email addresses for each user. For this reason, IAM Identity Center allows you to specify something other than an email address for user sign-in. IAM Identity Center requires that all user names and email addresses for your users are non-NULL and unique.

Groups

Groups are a logical combination of users that you define. You can create groups and add users to the groups. IAM Identity Center does not support adding a group to a group (nested groups). Groups are useful when assigning access to AWS accounts and applications. Rather than assign each user individually, you give permissions to a group. Later, as you add or remove users from a group, the user dynamically gets or loses access to accounts and applications that you assigned to the group.

User and group provisioning

Provisioning is the process of making user and group information available for use by IAM Identity Center and AWS managed applications or customer managed applications. You can create users and groups directly in IAM Identity Center, or work with users and groups you have in Active Directory or an external identity provider. Before you can use IAM Identity Center to assign users and groups access permissions in an AWS account, IAM Identity Center must be aware of the users and groups. Similarly, AWS managed applications and customer managed applications can work with users and groups of which IAM Identity Center is aware.

Provisioning in IAM Identity Center varies based on the identity source that you use. For more information, see [Manage your identity source](#).

Manage your identity source

Your identity source in IAM Identity Center defines where your users and groups are managed. After you configure your identity source, you can look up users or groups to grant them single sign-on access to AWS accounts applications, or both.

You can have only one identity source per organization in AWS Organizations. You can choose one of the following as your identity source:

- **External identity provider** – Choose this option if you want to manage users in an external identity provider (IdP) such as Okta or Microsoft Entra ID.
- **Active Directory** – Choose this option if you want to continue managing users in either your AWS Managed Microsoft AD directory using AWS Directory Service or your self-managed directory in Active Directory (AD).
- **Identity Center directory** – When you enable IAM Identity Center for the first time, it's automatically configured with an Identity Center directory as your default identity source unless you choose a different identity source. With the Identity Center directory, you create your users and groups, and assign their level of access to your AWS accounts and applications.

Note

IAM Identity Center does not support SAMBA4-based Simple AD as an identity source.

Topics

- [Considerations for changing your identity source](#)
- [Change your identity source](#)
- [Manage sign-in and attribute use for all identity source types](#)
- [Manage identities in IAM Identity Center](#)
- [Connect to a Microsoft AD directory](#)
- [Manage an external identity provider](#)

Considerations for changing your identity source

Although you can change your identity source at any time, we recommend that you consider how this change might affect your current deployment.

If you're already managing users and groups in one identity source, changing to a different identity source might remove all user and group assignments that you configured in IAM Identity Center. If this occurs, all users, including the administrative user in IAM Identity Center, will lose single sign-on access to their AWS accounts and applications.

Before you change the identity source for IAM Identity Center, review the following considerations before you proceed. If you want to proceed with changing your identity source, see [Change your identity source](#) for more information.

Changing between IAM Identity Center and Active Directory

If you're already managing users and groups in Active Directory, we recommend that you consider connecting your directory when you enable IAM Identity Center and choose your identity source. Do this before you create any users and groups in the default Identity Center directory and make any assignments.

If you're already managing users and groups in the default Identity Center directory, consider the following:

- **Assignments removed and users and groups deleted** – Changing your identity source to Active Directory deletes your users and groups from the Identity Center directory. This change also removes your assignments. In this case, after you change to Active Directory, you must synchronize your users and groups from Active Directory into the Identity Center directory, and then reapply their assignments.

If you choose to not use Active Directory, you must create your users and groups in the Identity Center directory, and then make assignments.

- **Assignments aren't deleted when identities are deleted** – When identities are deleted in the Identity Center directory, corresponding assignments also get deleted in IAM Identity Center. However in Active Directory, when identities are deleted (either in Active Directory or the synced identities), corresponding assignments aren't deleted.
- **No outbound synchronization for APIs** – If you use Active Directory as your identity source, we recommend that you use the [Create, Update, and Delete](#) APIs with caution. IAM Identity Center doesn't support outbound synchronization, so your identity source doesn't automatically update with the changes that you make to users or groups using these APIs.
- **Access portal URL will change** – Changing your identity source between IAM Identity Center and Active Directory also changes the URL for the AWS access portal.
- **Existing user session expiration can take up to two hours** – Once the users and groups are deleted from the Identity Center directory, users with active sessions can continue to access the AWS access portal and integrated AWS applications for up to two hours. For information about authentication session duration and user behavior, see [Authentication in IAM Identity Center](#).

For information about how IAM Identity Center provisions users and groups, see [Connect to a Microsoft AD directory](#).

Changing from IAM Identity Center to an external IdP

If you change your identity source from IAM Identity Center to an external identity provider (IdP), consider the following:

- **Assignments and memberships work with correct assertions** – your user assignments, group assignments, and group memberships continue to work as long as the new IdP sends the correct assertions (for example, SAML nameIDs). These assertions must match the user names and groups in IAM Identity Center.
- **No outbound synchronization** – IAM Identity Center doesn't support outbound synchronization, so your external IdP won't automatically update with changes to users and groups that you make in IAM Identity Center.
- **SCIM provisioning** – if you are using SCIM provisioning, changes to users and groups in your identity provider reflect only in IAM Identity Center after your identity provider sends those changes to IAM Identity Center. See [Considerations for using automatic provisioning](#).

- **Rollback** – you can revert your identity source back to using IAM Identity Center at any time. See [Changing from an external IdP to IAM Identity Center](#).
- **Existing user sessions are revoked on session duration expiry** – Once you change your identity source to an external identity provider, active user sessions persist for the remainder of the maximum session duration configured in the console. For example, if the AWS access portal session duration is set to eight hours, and you changed the identity source in the fourth hour, active user sessions persist for an additional four hours. To revoke user sessions, see [Delete sessions for the AWS access portal and AWS integrated applications](#).

If users are deleted or disabled in the IAM Identity Center console, using Identity Store APIs, or SCIM provisioning, users with active sessions can continue to access the AWS access portal and integrated AWS applications for up to two hours.

 **Note**

You won't be able to revoke user sessions from the IAM Identity Center console after you've deleted the user.


For information about how IAM Identity Center provisions users and groups, see [Manage an external identity provider](#).

Changing from an external IdP to IAM Identity Center

If you change your identity source from an external identity provider (IdP) to IAM Identity Center, consider the following:

- IAM Identity Center preserves all your assignments.
- **Force password reset** – Users who had passwords in IAM Identity Center can continue signing in with their old passwords. For users who were in the external IdP and weren't in IAM Identity Center, an administrator must force a password reset.
- **Existing user sessions are revoked on session duration expiry** – Once you change your identity source to IAM Identity Center, active user sessions persist for the remaining duration of the maximum session duration configured in the console. For example, if the AWS access portal session duration is eight hours, and you changed the identity source at fourth hour, active user sessions continue to run for an additional four hours. To revoke user sessions, see [Delete sessions for the AWS access portal and AWS integrated applications](#).

If users are deleted or disabled in the IAM Identity Center console, using Identity Store APIs, or SCIM provisioning, users with active sessions can continue to access the AWS access portal and integrated AWS applications for up to two hours.

 **Note**

You won't be able to revoke user sessions from the IAM Identity Center console after you've deleted the user.

For information about how IAM Identity Center provisions users and groups, see [Manage identities in IAM Identity Center](#).

Changing from one external IdP to another external IdP


If you're already using an external IdP as your identity source for IAM Identity Center and you change to a different external IdP, consider the following:

- **Assignments and memberships work with correct assertions** – IAM Identity Center preserves all of your assignments. The user assignments, group assignments, and group memberships continue to work as long as the new IdP sends the correct assertions (for example, SAML nameIDs).

These assertions must match the user names in IAM Identity Center when your users authenticate through the new external IdP.

- **SCIM provisioning** – If you are using SCIM for provisioning into IAM Identity Center, we recommend that you review the IdP-specific information in this guide and the documentation provided by the IdP to ensure that the new provider matches users and groups correctly when SCIM is enabled.
- **Existing user sessions are revoked on session duration expiry** – Once you change your identity source to different external identity provider, active user sessions persist for the remaining duration of the maximum session duration configured in the console. For example, if the AWS access portal session duration is eight hours, and you changed the identity source at fourth hour, active user sessions persist for an additional four hours. To revoke user sessions, see [Delete sessions for the AWS access portal and AWS integrated applications](#).

If users are deleted or disabled in the IAM Identity Center console, using Identity Store APIs, or SCIM provisioning, users with active sessions can continue to access the AWS access portal and integrated AWS applications for up to two hours.

 **Note**

You won't be able to revoke user sessions from the IAM Identity Center console after you've deleted the user.

For information about how IAM Identity Center provisions users and groups, see [Manage an external identity provider](#).

Changing between Active Directory and an external IdP

If you change your identity source from an external IdP to Active Directory, or from Active Directory to an external IdP, consider the following:

- **Users, groups, and assignments are deleted** – All users, groups, and assignments are deleted from IAM Identity Center. No user or group information is affected in either the external IdP or Active Directory.
- **Provisioning users** – If you change to an external IdP, you must configure IAM Identity Center to provision your users. Alternatively, you must manually provision the users and groups for the external IdP before you can configure assignments.
- **Create assignments and groups** – If you change to Active Directory, you must create assignments with the users and groups that are in your directory in Active Directory.
- **Existing user sessions expiration can take up to two hours** – Once the users and groups are deleted from the Identity Center directory, users with active sessions can continue to access the AWS access portal and integrated AWS applications for up to two hours. For information about authentication session duration and user behavior, see [Authentication in IAM Identity Center](#).

For information about how IAM Identity Center provisions users and groups, see [Connect to a Microsoft AD directory](#).

Change your identity source

The following procedure describes how to change from a directory that IAM Identity Center provides (the default Identity Center directory) to Active Directory or an external identity provider, or the other way around. Before you proceed, review the information in [Considerations for changing your identity source](#). Depending on your current deployment, this change might remove any user and group assignments that you configured in IAM Identity Center. If this occurs, all users, including the administrative user in IAM Identity Center, will lose single sign-on access to their AWS accounts and applications.

To change your identity source

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab. Choose **Actions**, and then choose **Change identity source**.
4. Under **Choose identity source**, select the source that you want to change to, and then choose **Next**.

If you are changing to Active Directory, choose the available directory from the menu on the next page.

Important

Changing your identity source to or from Active Directory deletes users and groups from the Identity Center directory. This change also removes any assignments that you configured in IAM Identity Center.

If you are switching to an external identity provider, we recommend that you follow the steps in [How to connect to an external identity provider](#).

5. After you read the disclaimer and are ready to proceed, type **ACCEPT**.
6. Choose **Change identity source**. If you are changing your identity source to Active Directory, proceed to the next step.
7. Changing your identity source to Active Directory takes you to the **Settings** page. On the **Settings** page, do either of the following:

- Choose **Start guided setup**. For information about how to complete the guided setup process, see [Guided setup](#).
- In the **Identity source** section, choose **Actions**, and then choose **Manage sync** to configure your *sync scope*, the list of users and groups to sync.

Manage sign-in and attribute use for all identity source types

IAM Identity Center provides the following set of features that enables admins to control AWS access portal use, to set session durations for users in the AWS access portal and your applications, and to use attributes for access control. These features work with an Identity Center directory or external identity provider as your identity source.

Note

If you're using Active Directory as an identity source for IAM Identity Center, session management isn't supported.

Topics

- [Configure the session duration of the AWS access portal and IAM Identity Center integrated applications](#)
- [Delete sessions for the AWS access portal and AWS integrated applications](#)
- [Supported user and group attributes](#)

Configure the session duration of the AWS access portal and IAM Identity Center integrated applications

The IAM Identity Center administrator can configure the session duration for both applications integrated with IAM Identity Center and the AWS access portal. The session duration of authentication into the AWS access portal and IAM Identity Center integrated applications is the maximum length of time that a user can be signed in without re-authenticating. The IAM Identity Center administrator can end an active AWS access portal session and by doing so also end the sessions of integrated applications.

The default session duration is 8 hours. The IAM Identity Center administrator can specify a different duration, from a minimum of 15 minutes to a maximum of 90 days. For more information about authentication session duration and user behavior, see [Authentication in IAM Identity Center](#).

Note

Modifying the AWS access portal session duration and ending the AWS access portal sessions have no effect on the AWS Management Console session duration that you define in your permission sets.

The following topics provide information about configuring the session duration of the AWS access portal and IAM Identity Center integrated applications.

Topics

- [Prerequisites and considerations](#)
- [How to configure application session duration](#)
- [How to extend the session duration for Amazon Q Developer](#)

Prerequisites and considerations

The following are the prerequisites and considerations for configuring the session duration for the AWS access portal and IAM Identity Center integrated applications.

External identity providers

IAM Identity Center uses `SessionNotOnOrAfter` attribute from SAML assertions to help determine how long the session can be valid for.

- If `SessionNotOnOrAfter` is not passed in a SAML assertion, the duration of an AWS access portal session is not impacted by the duration of your external IdP session. For example, if your IdP session duration is 24 hours and you set an 18-hour session duration in IAM Identity Center, your users must re-authenticate in the AWS access portal after 18 hours.
- If `SessionNotOnOrAfter` is passed in a SAML assertion, the session duration value is set to the shorter of the AWS access portal session duration and your SAML IdP session duration. If you set a 72-hour session duration in IAM Identity Center and your IdP has a session duration of 18 hours, your users will have access to AWS resources for the 18 hours defined in your IdP.

- If the session duration of your IdP is longer than the one set in IAM Identity Center, your users will be able to start a new IAM Identity Center session without re-entering their credentials, based on their still-valid login session with your IdP.

Note

If you're using Active Directory as an identity source for IAM Identity Center, session management isn't supported.

AWS CLI and SDK sessions

If you're using the AWS Command Line Interface, AWS Software Development Kits (SDKs), or other AWS development tools to access AWS services programmatically, the following prerequisites must be met to set session duration for the AWS access portal and the IAM Identity Center integrated applications.

- You must [configure the AWS access portal session duration](#) in the IAM Identity Center console.
- You must define a profile for single sign-on settings in your shared AWS config file. This profile is used to connect to the AWS access portal. We recommend that you use the SSO token provider configuration. With this configuration, your AWS SDK or tool can automatically retrieve refreshed authentication tokens. For more information, see [SSO token provider configuration](#) in the *AWS SDK and Tools Reference Guide*.
- Users must run a version of the AWS CLI or an SDK that supports session management.

Minimum versions of the AWS CLI that support session management

Following are the minimum versions of the AWS CLI that support session management.

- AWS CLI V2 2.9 or later
- AWS CLI V1 1.27.10 or later

For information about how to install or update the latest AWS CLI version, see [Installing or updating the latest version of the AWS CLI](#).

If your users are running the AWS CLI, if you refresh your permission set just before the IAM Identity Center session is set to expire and the session duration is set to 20 hours while the

permission set duration is set to 12 hours, the AWS CLI session runs for the maximum of 20 hours plus 12 hours for a total of 32 hours. For more information about the IAM Identity Center CLI, see [AWS CLI Command Reference](#).

Minimum versions of SDKs that support IAM Identity Center session management

Following are the minimum versions of the SDKs that support IAM Identity Center session management.

SDK	Minimum version
Python	1.26.10
PHP	3.245.0
Ruby	aws-sdk-core 3.167.0
Java V2	AWS SDK for Java v2 (2.18.13)
Go V2	Whole SDK: release-2022-11-11 and specific Go modules: credentials/v1.13.0, config/v1.18.0
JS V2	2.1253.0
JS V3	v3.210.0
C++	1.9.372
.NET	v3.7.400.0

How to configure application session duration

Use the following procedure to configure the session duration of the AWS access portal and IAM Identity Center integrated applications.

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.

4. Under **Authentication**, next to **Session settings**, choose **Configure**. A **Configure session settings** dialog box appears.
5. In the **Configure session settings** dialog box, choose the maximum session duration in minutes, hours, and days for your users by selecting the drop down arrow. Choose a the length for the session, and then choose **Save**. You return to the **Settings** page.

How to extend the session duration for Amazon Q Developer

If your developers use Amazon Q Developer as part of an integrated development environment (IDE), you can set the session duration for Amazon Q Developer to 90 days. Depending on when you enabled IAM Identity Center, extended session duration for Amazon Q Developer might be enabled by default. This extended session doesn't affect the session duration of the AWS access portal or other IAM Identity Center integrated applications.

Note

Amazon Q Developer is accessible from consoles set to commercial AWS Regions that are enabled by default. If your IAM Identity Center instance is located in a Region where Amazon Q Developer isn't currently accessible, enabling 90 day extended session duration won't override the default setting. This means that your session duration remains unchanged, regardless of whether you enable 90 day extended session duration or not. For information, see [Supported Regions for Amazon Q Developer](#).

Enable or disable 90 day extended session duration for Amazon Q Developer.

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. Under **Authentication**, next to **Session settings**, choose **Configure**. A **Configure session settings** dialog box appears.
5. In the **Configure session settings** dialog box, select the checkbox for **Enable extended sessions for Amazon Q Developer**. Deselect the checkbox to disable extended session duration.
6. Choose **Save** to return to the **Settings** page.

Delete sessions for the AWS access portal and AWS integrated applications

Use the following procedure to view and delete active sessions for an IAM Identity Center user.

Note

Deleting an active session for an IAM Identity Center user does not delete any active IAM Role sessions in the AWS Management Console or AWS CLI.

To delete an active session of the AWS access portal and IAM Identity Center integrated applications

1. Open the [IAM Identity Center console](#).
2. Choose **Users**.
3. On the **Users** page, choose the username of the user whose sessions you want to manage. This takes you to a page with the user's information.
4. On the user's page, choose the **Active sessions** tab. The number in parentheses next to **Active sessions** indicates the number of current active sessions for this user.
5. Select the check boxes beside the sessions that you want to delete, and then choose **Delete session**. A dialog box appears that confirms you're deleting active sessions for this user. Read the information in the dialog box, and if you want to continue, choose **Delete session**.
6. You are returned to the user's page. A green flash bar appears to indicate that the selected sessions were successfully deleted.

For more information on the behavior of revoked authentication sessions, see [Authentication sessions](#).

Supported user and group attributes

Attributes are pieces of information that help you define and identify individual user or group objects, such as name, email, or members. IAM Identity Center supports most commonly used attributes regardless if they are entered manually during user creation or when automatically provisioned using a synchronization engine such as defined in the System for Cross-Domain Identity Management (SCIM) specification. For more information about this specification, see <https://tools.ietf.org/html/rfc7642>. For more information about manual and automatic provisioning, see [Provisioning when users come from an external IdP](#).

Because IAM Identity Center supports SCIM for automatic provisioning use cases, the Identity Center directory supports all of the same user and group attributes that are listed in the SCIM specification, with a few exceptions. The following sections describe which attributes are not supported by IAM Identity Center.

User objects

All attributes from the SCIM user schema (<https://tools.ietf.org/html/rfc7643#section-8.3>) are supported in the IAM Identity Center identity store, except for the following:

- password
- ims
- photos
- entitlements
- x509Certificates

All sub-attributes for users are supported, except for the following:

- 'display' sub-attribute of any multi-valued attribute (For example, emails or phoneNumbers)
- 'version' sub-attribute of 'meta' attribute

Group objects

All attributes from the SCIM group schema (<https://tools.ietf.org/html/rfc7643#section-8.4>) are supported.

All sub-attributes for groups are supported, except for the following:

- 'display' sub-attribute of any multi-valued attribute (For example, members).

Manage identities in IAM Identity Center

IAM Identity Center provides the following capabilities for your users and groups:

- Create your users and groups.
- Add your users as members to the groups.

- Assign the groups with the desired level of access to your AWS accounts and applications.

To manage users and groups in the IAM Identity Center store, AWS supports the API operations listed in [Identity Center Actions](#).

Provisioning when users are in IAM Identity Center

When you create users and groups directly in IAM Identity Center, provisioning is automatic. These identities are immediately available for use in making assignments and for use by applications. For more information, see [User and group provisioning](#).

Changing your identity source

If you prefer to manage users in AWS Managed Microsoft AD, you can stop using your Identity Center directory at any time and instead connect IAM Identity Center to your directory in Microsoft AD by using AWS Directory Service. For more information, see considerations for [Changing between IAM Identity Center and Active Directory](#).

If you prefer to manage users in an external identity provider (IdP), you can connect IAM Identity Center to your IdP and enable automatic provisioning. For more information, see considerations for [Changing from IAM Identity Center to an external IdP](#).

Topics

- [Add users to your Identity Center directory](#)
- [Add groups to your Identity Center directory](#)
- [Add users to groups](#)
- [Delete groups in IAM Identity Center](#)
- [Delete users in IAM Identity Center](#)
- [Disable user access to AWS accounts and applications in IAM Identity Center](#)
- [Edit Identity Center directory user properties](#)
- [Reset the IAM Identity Center user password for an end user](#)
- [Email one-time password to users created with API](#)
- [Password requirements when managing identities in IAM Identity Center](#)

Add users to your Identity Center directory

Users and groups that you create in your Identity Center directory are available in IAM Identity Center only. Use the following procedure to add users to your Identity Center directory using the IAM Identity Center console. Alternatively, you can call the AWS API operation [CreateUser](#) to add users.

To add a user

1. Open the [IAM Identity Center console](#).
2. Choose **Users**.
3. Choose **Add user** and provide the following required information:
 - a. **Username** – This user name is required to sign in to the AWS access portal and can't be changed later. It must be between 1 and 100 characters.
 - b. **Password** – You can either send an email with the password setup instructions (this is the default option) or generate a one-time password. If you are creating an administrative user and you choose to send an email, make sure that you specify an email address that you can access.
 - i. **Send an email to this user with password setup instructions.** – This option automatically sends the user an email addressed from Amazon Web Services, with the subject line **Invitation to join AWS IAM Identity Center**. The email invites the user on behalf of your company to access the IAM Identity Center AWS access portal, and registers a password. The email invitation will expire in seven days. If this happens, you can resend the email by choosing **Reset password**, and then choosing **Send an email to the user with instructions for resetting the password**. Before the user accepts the invitation, you will see **Send email verification link**, which is meant to verify their email address. However, this step is optional and will disappear after the user accepts the invitation and registers a password.


Note

In certain Regions, IAM Identity Center sends emails to users using Amazon Simple Email Service from another AWS Region. For information about how emails are sent, see [Cross-Region calls](#).

All emails sent by the IAM Identity Center service will come from either the address `no-reply@signin.aws.com` or `no-reply@login.awsapps.com`.

We recommend that you configure your email system so that it accepts emails from these sender email addresses and does not handle them as junk or spam.

- ii. **Generate a one-time password that you can share with this user.** – This option provides you with the AWS access portal URL and password details that you can manually send to the user from your email address. The user will need to verify their email address. You can initiate the process by choosing **Send email verification link**. The email verification link will expire in seven days. If this happens, you can resend the email verification link by choosing **Reset password**, and then choosing **Generate a one-time password and share the password with the user**.
- c. **Email address** – The email address must be unique.
- d. **Confirm email address**
- e. **First name** – You must enter a name here for automatic provisioning to work. For more information, see [Provisioning an external identity provider into IAM Identity Center using SCIM](#).
- f. **Last name** – You must enter a name here for automatic provisioning to work.
- g. **Display name**

 **Note**

(Optional) If applicable, you can specify values for additional attributes such as the user's **Microsoft 365 immutable ID** to help provide the user with single sign-on access to certain business applications.

4. Choose **Next**.
5. If applicable, select one or more groups to which you want to add the user, and choose **Next**.
6. Review the information that you specified for **Step 1: Specify user details** and **Step 2: Add user to groups - optional**. Choose **Edit** by either step to make any changes. After you confirm that the correct information is specified for both steps, choose **Add user**.

Add groups to your Identity Center directory

Use the following procedure to add groups to your Identity Center directory using the IAM Identity Center console. Alternatively, you can call the AWS API operation [CreateGroup](#) to add groups.

To add a group

1. Open the [IAM Identity Center console](#).
2. Choose **Groups**.
3. Choose **Create group**.
4. Enter a **Group name** and **Description - optional**. The description should provide details on what permissions have been or will be assigned to the group. Under **Add users to group - optional**, locate the users you want to add as members. Then select the check box next to each of them.
5. Choose **Create group**.

After you add this group to your Identity Center directory, you can assign single sign-on access to this group. For more information, see [Assign user access to AWS accounts](#).

Add users to groups

Use the following procedure to add users as members of a group that you previously created in your Identity Center directory using the IAM Identity Center console. Alternatively, you can call the AWS API operation [CreateGroupMembership](#) to add a user as a member of a group.

To add a user as a member of a group

1. Open the [IAM Identity Center console](#).
2. Choose **Groups**.
3. Choose the **group name** that you want to update.
4. On the group details page, under **Users in this group**, choose **Add users to group**.
5. On the **Add users to group** page, under **Other users**, locate the users you want to add as members. Then, select the check box next to each of them.
6. Choose **Add users**.

Delete groups in IAM Identity Center

When you delete a group in your IAM Identity Center directory, it removes access to AWS accounts and applications for all users who are members of this group. After a group is deleted it cannot be undone. Use the following procedure to delete a group in your Identity Center directory using the IAM Identity Center console.

To delete a group in IAM Identity Center

Important

The instructions on this page apply to [AWS IAM Identity Center](#). They do not apply to [AWS Identity and Access Management \(IAM\)](#). IAM Identity Center users, groups, and user credentials are different from IAM users, groups, and IAM user credentials. If you are looking for instructions on deleting groups in IAM, see [Deleting an IAM user group](#) in the *AWS Identity and Access Management User Guide*.

1. Open the [IAM Identity Center console](#).
2. Choose **Groups**.
3. There are two ways you can delete a group:
 - On the **Groups** page, you can select multiple groups for deletion. Select the group name that you want to delete and choose **Delete group**.
 - Choose the group name that you want to delete. On the group details page, choose **Delete group**.
4. You might be asked to confirm your intent to delete the group.
 - If you delete multiple groups at once, confirm your intent by typing **Delete** in the **Delete group** dialog box.
 - If you delete a single group that contains users, confirm your intent by typing the name of the group you want to delete in the **Delete group** dialog box.
5. Choose **Delete group**. If you selected multiple groups for deletion, choose **Delete # groups**.

Delete users in IAM Identity Center

When you delete a user in your IAM Identity Center directory, it removes their access to AWS accounts and applications. After a user is deleted it cannot be undone. Use the following procedure to delete a user in your Identity Center directory using the IAM Identity Center console.

Note

When you disable user access or delete a user in IAM Identity Center, that user will immediately be prevented from signing in to the AWS access portal and will not be able to create new sign in sessions. For more information, see [Authentication sessions](#).

To delete a user in IAM Identity Center**Important**


The instructions on this page apply to [AWS IAM Identity Center](#). They do not apply to [AWS Identity and Access Management \(IAM\)](#). IAM Identity Center users, groups, and user credentials are different from IAM users, groups, and IAM user credentials. If you are looking for instructions on deleting users in IAM, see [Deleting an IAM user](#) in the *AWS Identity and Access Management User Guide*.

1. Open the [IAM Identity Center console](#).
2. Choose **Users**.
3. There are two ways you can delete a user:
 - On the **Users** page, you can select multiple users for deletion. Select the username that you want to delete and choose **Delete users**.
 - Choose the username that you want to delete. On the user details page, choose **Delete user**.
4. If you delete multiple users at once, confirm your intent by typing **Delete** in the **Delete user** dialog box.
5. Choose **Delete user**. If you selected multiple users for deletion, choose **Delete # users**.

Disable user access to AWS accounts and applications in IAM Identity Center

When you disable user access in your IAM Identity Center directory, you can't edit their user details, reset their password, add the user to a group, or view their group membership. Disabling user access prevents them from signing in to the AWS access portal and they will no longer have access to their assigned AWS accounts and applications.

Use the following procedure to disable user access in your Identity Center directory using the IAM Identity Center console.

 **Note**

When you disable user access or delete a user in IAM Identity Center, that user will immediately be prevented from signing in to the AWS access portal and won't be able to create new sign in sessions. For more information, see [Authentication sessions](#).

To disable user access in IAM Identity Center

1. Open the [IAM Identity Center console](#).

 **Important**

The instructions on this page apply to [AWS IAM Identity Center](#). They don't apply to [AWS Identity and Access Management](#) (IAM). IAM Identity Center users, groups, and user credentials are different from IAM users, groups, and IAM user credentials. If you are looking for instructions on deactivating users in IAM, see [Managing IAM users](#) in the *AWS Identity and Access Management User Guide*.

2. Choose **Users**.
3. Select the username of the user whose access you want to disable.
4. Below the username of the user whose access you want to disable, in the **General information** section, choose **Disable user access**.
5. In the **Disable user access** dialog box, choose **Disable user access**.

Edit Identity Center directory user properties

Use the following procedure to edit the properties of a user in your Identity Center directory using the IAM Identity Center console. Alternatively, you can call the AWS API operation [UpdateUser](#) to update user properties.

To edit user properties in IAM Identity Center

1. Open the [IAM Identity Center console](#).
2. Choose **Users**.

3. Choose the user that you want to edit.
4. On the user **Profile** page, next to **Profile details**, choose **Edit**.
5. On the **Edit profile details** page, update the properties as needed. Then, choose **Save changes**.

Note

(Optional) You can modify additional attributes such as **Employee number** and **Office 365 Immutable ID** to help map the user's identity in IAM Identity Center with certain business applications that users need to use.

Note

The **Email address** attribute is an editable field and the value you provide must be unique.

Reset the IAM Identity Center user password for an end user

This procedure is for administrators who need to reset the password for a user in your IAM Identity Center directory. You'll use the IAM Identity Center console to reset passwords.

Considerations for identity providers and user types

- **Microsoft Active Directory or external provider** – If you're connecting IAM Identity Center to Microsoft Active Directory or an external provider, user password resets must be done from within Active Directory or the external provider. This means that passwords for those users can't be reset from the IAM Identity Center console.
- **Users in the IAM Identity Center directory** – If you're an IAM Identity Center user, you can reset your own IAM Identity Center password, see [Resetting your AWS access portal user password](#).

To reset a password for an IAM Identity Center end user

Important

The instructions on this page apply to [AWS IAM Identity Center](#). They do not apply to [AWS Identity and Access Management](#) (IAM). IAM Identity Center users, groups, and user credentials are different from IAM users, groups, and IAM user credentials. If you are looking for instructions on changing passwords for IAM users, see [Managing passwords for IAM users](#) in the *AWS Identity and Access Management User Guide*.

1. Open the [IAM Identity Center console](#).
2. Choose **Users**.
3. Select the username of the user whose password you want to reset.
4. On the user details page, choose **Reset password**.
5. In the **Reset password** dialog box, select one of the following choices, and then choose **Reset password**:
 - a. **Send an email to the user with instructions to reset the password** – This option automatically sends the user an email addressed from Amazon Web Services that walks them through how to reset their password.

Warning

As a security best practice, verify that the email address for this user is correct prior to selecting this option. If this password reset email were to be sent to an incorrect or misconfigured email address, a malicious recipient could use it to gain unauthorized access to your AWS environment.

- b. **Generate a one-time password and share the password with the user** – This option provides you with the password details that you can manually send to the user from your email address.

Email one-time password to users created with API

When you create users with the [CreateUser](#) API operation, they don't have passwords.

You can choose to send users created with the CreateUser API an email with a one-time password (OTP) after their first attempt to sign in, if you've specified an email for the user when they were created. After receiving the email OTP, when a user signs in, they must set a new password. If you don't enable this setting, then you must [generate a one-time password and share](#) with users that you create using the CreateUser API.

To send an email OTP to users created with the CreateUser API

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. In the **Standard authentication** section, choose **Configure**.
5. A dialog box appears. Check the box next to **Send email OTP**. Then, choose **Save**. The status updates from **Disabled** to **Enabled**.

Password requirements when managing identities in IAM Identity Center

Note

These requirements apply only to users created in the Identity Center directory. If you have configured an identity source other than IAM Identity Center for authentication, such as [Active Directory](#) or an [external identity provider](#), the password policies for your users are defined and enforced in those systems, not in IAM Identity Center. If your identity source is AWS Managed Microsoft AD, see [Manage password policies for AWS Managed Microsoft AD](#) for more information.

When you use IAM Identity Center as your identity source, users must adhere to the following password requirements to set or change their password:

- Passwords are case-sensitive.
- Passwords must be between 8 and 64 characters in length.
- Passwords must contain at least one character from each of the following four categories:
 - Lowercase letters (a-z)
 - Uppercase letters (A-Z)
 - Numbers (0-9)

- Non-alphanumeric characters (~!@#%&* _+= ` \()\{\}[];:"'<>,.?/)
- The last three passwords cannot be reused.
- Passwords that are publicly known through a data set leaked from a third party cannot be used.

Connect to a Microsoft AD directory

With AWS IAM Identity Center, you can connect a self-managed directory in Active Directory (AD) or a directory in AWS Managed Microsoft AD by using AWS Directory Service. This Microsoft AD directory defines the pool of identities that administrators can pull from when using the IAM Identity Center console to assign single sign-on access. After connecting your corporate directory to IAM Identity Center, you can then grant your AD users or groups access to AWS accounts, applications, or both.

AWS Directory Service helps you to set up and run a standalone AWS Managed Microsoft AD directory hosted in the AWS Cloud. You can also use AWS Directory Service to connect your AWS resources with an existing self-managed AD. To configure AWS Directory Service to work with your self-managed AD, you must first set up trust relationships to extend authentication to the cloud.

IAM Identity Center uses the connection provided by AWS Directory Service to perform pass-through authentication to the source AD instance. When you use AWS Managed Microsoft AD as your identity source, IAM Identity Center can work with users from AWS Managed Microsoft AD or from any domain connected through an AD trust. If you want to locate your users in four or more domains, users must use the `DOMAIN\user` syntax as their user name when performing sign-ins to IAM Identity Center.

Notes

- As a prerequisite step, make sure your AD Connector or directory in AWS Managed Microsoft AD in AWS Directory Service resides within your AWS Organizations management account. For more information, see [Confirm your identity sources in IAM Identity Center](#).
- IAM Identity Center does not support SAMBA 4-based Simple AD as a connected directory.

Considerations for using Active Directory

If you want to use Active Directory as your identity source, your configuration must meet the following prerequisites:

- If you're using AWS Managed Microsoft AD, you must enable IAM Identity Center in the same AWS Region where your AWS Managed Microsoft AD directory is set up. IAM Identity Center stores the assignment data in the same Region as the directory. To administer IAM Identity Center, you might need to switch to the Region where IAM Identity Center is configured. Also, note that the AWS access portal uses the same access URL as your directory.
- Use an Active Directory residing in the management account:

You must have an existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory Service, and it must reside within your AWS Organizations management account. You can connect only one AD Connector directory or one directory in AWS Managed Microsoft AD at a time. If you need to support multiple domains or forests, use AWS Managed Microsoft AD. For more information, see:

- [Connect a directory in AWS Managed Microsoft AD to IAM Identity Center](#)
- [Connect a self-managed directory in Active Directory to IAM Identity Center](#)
- Use an Active Directory residing in the delegated admin account:

If you plan to enable IAM Identity Center delegated admin and use Active Directory as your IAM Identity Center identity source, you can use an existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory residing in the delegated admin account.


If you decide to change IAM Identity Center identity source from any other source to Active Directory, or change it from Active Directory to any other source, the directory must reside in (be owned by) the IAM Identity Center delegated administrator member account if one exists; otherwise, it must be in the management account.

Connect Active Directory and specify a user

If you're already using Active Directory, the following topics will help you prepare to connect your directory to IAM Identity Center.

You can connect an AWS Managed Microsoft AD directory or a self-managed directory in Active Directory with IAM Identity Center. If you plan to connect an AWS Managed Microsoft AD directory

or a self-managed directory in Active Directory, make sure that your Active Directory configuration meets the prerequisites in [Confirm your identity sources in IAM Identity Center](#).

 **Note**

As a security best practice, we strongly recommend that you enable multi-factor authentication. If you plan to connect an AWS Managed Microsoft AD directory or a self-managed directory in Active Directory and you're not using RADIUS MFA with AWS Directory Service, enable MFA in IAM Identity Center.

AWS Managed Microsoft AD

1. Review the guidance in [Connect to a Microsoft AD directory](#).
2. Follow the steps in [Connect a directory in AWS Managed Microsoft AD to IAM Identity Center](#).
3. Configure Active Directory to synchronize the user to whom you want to grant administrative permissions into IAM Identity Center. For more information, see [Synchronize an administrative user into IAM Identity Center](#).

Self-managed directory in Active Directory

1. Review the guidance in [Connect to a Microsoft AD directory](#).
2. Follow the steps in [Connect a self-managed directory in Active Directory to IAM Identity Center](#).
3. Configure Active Directory to synchronize the user to whom you want to grant administrative permissions into IAM Identity Center. For more information, see [Synchronize an administrative user into IAM Identity Center](#).

External IdP

1. Review the guidance in [Manage an external identity provider](#).
2. Follow the steps in [How to connect to an external identity provider](#).
3. Configure your IdP to provision users into IAM Identity Center.

Note

Before you set up automatic, group-based provisioning of all your workforce identities from your IdP into IAM Identity Center, we recommend that you sync the one user to whom you want to grant administrative permissions into IAM Identity Center.

Synchronize an administrative user into IAM Identity Center

After you connect your directory to IAM Identity Center, you can specify a user to whom you want to grant administrative permissions, and then synchronize that user from your directory into IAM Identity Center.

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. On the **Manage Sync** page, choose the **Users** tab, and then choose **Add users and groups**.
5. On the **Users** tab, under **User**, enter the exact user name and choose **Add**.
6. Under **Added Users and Groups**, do the following:
 - a. Confirm that the user to whom you want to grant administrative permissions is specified.
 - b. Select the check box to the left of the user name.
 - c. Choose **Submit**.
7. In the **Manage sync** page, the user that you specified appears in the **Users in sync scope** list.
8. In the navigation pane, choose **Users**.
9. On the **Users** page, it might take some time for the user that you specified to appear in the list. Choose the refresh icon to update the list of users.

At this point, your user doesn't have access to the management account. You will set up administrative access to this account by creating an administrative permission set and assigning the user to that permission set. For more information, see [Create a permission set for job functions](#).

Provisioning when users come from Active Directory

IAM Identity Center uses the connection provided by the AWS Directory Service to synchronize user, group, and membership information from your source directory in Active Directory to the IAM Identity Center identity store. No password information is synchronized to IAM Identity Center, because user authentication takes place directly from the source directory in Active Directory. This identity data is used by applications to facilitate in-app lookup, authorization, and collaboration scenarios without passing LDAP activity back to the source directory in Active Directory.

For more information about provisioning, see [User and group provisioning](#).

Topics

- [Connect a directory in AWS Managed Microsoft AD to IAM Identity Center](#)
- [Connect a self-managed directory in Active Directory to IAM Identity Center](#)
- [Attribute mappings for AWS Managed Microsoft AD directory](#)
- [Provision users and groups from Active Directory](#)

Connect a directory in AWS Managed Microsoft AD to IAM Identity Center

Use the following procedure to connect a directory in AWS Managed Microsoft AD that is managed by AWS Directory Service to IAM Identity Center.

To connect AWS Managed Microsoft AD to IAM Identity Center

1. Open the [IAM Identity Center console](#).

Note

Make sure that the IAM Identity Center console is using one of the Regions where your AWS Managed Microsoft AD directory is located before you move to the next step.

2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Change identity source**.
4. Under **Choose identity source**, select **Active Directory**, and then choose **Next**.
5. Under **Connect active directory**, choose a directory in AWS Managed Microsoft AD from the list, and then choose **Next**.

6. Under **Confirm change**, review the information and when ready type **ACCEPT**, and then choose **Change identity source**.

Important

To specify a user in Active Directory as an administrative user in IAM Identity Center, you must first synchronize the user to whom you want to grant administrative permissions from Active Directory into IAM Identity Center. To do so, follow the steps in [Synchronize an administrative user into IAM Identity Center](#).

Connect a self-managed directory in Active Directory to IAM Identity Center

Users in your self-managed directory in Active Directory (AD) can also have single sign-on access to AWS accounts and applications in the AWS access portal. To configure single sign-on access for these users, you can do either of the following:

- **Create a two-way trust relationship** – When two-way trust relationships are created between AWS Managed Microsoft AD and a self-managed directory in AD, users in your self-managed directory in AD can sign in with their corporate credentials to various AWS services and business applications. One-way trusts do not work with IAM Identity Center.

AWS IAM Identity Center requires a two-way trust so that it has permissions to read user and group information from your domain to synchronize user and group metadata. IAM Identity Center uses this metadata when assigning access to permission sets or applications. User and group metadata is also used by applications for collaboration, like when you share a dashboard with another user or group. The trust from AWS Directory Service for Microsoft Active Directory to your domain permits IAM Identity Center to trust your domain for authentication. The trust in the opposite direction grants AWS permissions to read user and group metadata.

For more information about setting up a two-way trust, see [When to Create a Trust Relationship](#) in the *AWS Directory Service Administration Guide*.

Note

In order to use AWS applications, like IAM Identity Center to read AWS Directory Service directory users from trusted domains, the AWS Directory Service accounts require permissions to the `userAccountControl` attribute on the trusted users. Without read

permissions to this attribute, AWS applications are unable to determine if the account is enabled or disabled.

Read access to this attribute is provided by default when a trust is created. If you deny access to this attribute (not recommended), you will break applications like Identity Center from being able to read trusted users. The solution is to specifically allow Read access to the `userAccountControl` attribute on the AWS service accounts under the AWS Reserved OU (prefixed with `AWS_`).

- **Create an AD Connector** – AD Connector is a directory gateway that can redirect directory requests to your self-managed AD without caching any information in the cloud. For more information, see [Connect to a Directory](#) in the *AWS Directory Service Administration Guide*. The following are considerations when using AD Connector:
 - If you are connecting IAM Identity Center to an AD Connector directory, any future user password resets must be done from within AD. This means that users will not be able to reset their passwords from the AWS access portal.
 - If you use AD Connector to connect your Active Directory Domain Service to IAM Identity Center, IAM Identity Center only has access to the users and groups of the single domain to which AD Connector attaches. If you need to support multiple domains or forests, use AWS Directory Service for Microsoft Active Directory.

Note

IAM Identity Center does not work with SAMBA4-based Simple AD directories.

Attribute mappings for AWS Managed Microsoft AD directory

Attribute mappings are used to map attribute types that exist in IAM Identity Center with like attributes in an AWS Managed Microsoft AD directory. IAM Identity Center retrieves user attributes from your Microsoft AD directory and maps them to IAM Identity Center user attributes. These IAM Identity Center user attribute mappings are also used for generating SAML 2.0 assertions for your applications. Each application determines the list of SAML 2.0 attributes it needs for successful single sign-on.

IAM Identity Center prefills a set of attributes for you under the **Attribute mappings** tab found on your application's configuration page. IAM Identity Center uses these user attributes to populate SAML assertions (as SAML attributes) that are sent to the application. These user attributes are in

turn retrieved from your Microsoft AD directory. For more information, see [Map attributes in your application to IAM Identity Center attributes](#).

IAM Identity Center also manages a set of attributes for you under the **Attribute mappings** section of your directory configuration page. For more information, see [Map attributes in IAM Identity Center to attributes in your AWS Managed Microsoft AD directory](#).

Supported directory attributes

The following table lists all AWS Managed Microsoft AD directory attributes that are supported and that can be mapped to user attributes in IAM Identity Center.

Supported attributes in your Microsoft AD directory

`${dir:email}`

`${dir:displayname}`

`${dir:distinguishedName}`

`${dir:firstname}`

`${dir:guid}`

`${dir:initials}`

`${dir:lastname}`

`${dir:proxyAddresses}`

`${dir:proxyAddresses:smtp}`

`${dir:proxyAddresses:SMTP}`

`${dir:windowsUpn}`

You can specify any combination of supported Microsoft AD directory attributes to map to a single mutable attribute in IAM Identity Center. For example, you can choose the subject attribute under the **User attribute in IAM Identity Center** column. Then map it to either `${dir:displayname}` or `${dir:lastname}${dir:firstname }` or any single supported

attribute or any arbitrary combination of supported attributes. For a list of the default mappings for user attributes in IAM Identity Center, see [Default mappings](#).

Warning

Certain IAM Identity Center attributes can't be modified because they are immutable and mapped by default to specific Microsoft AD directory attributes. For example, "username" is a mandatory attribute in IAM Identity Center. If you map "username" to an AD directory attribute with an empty value, IAM Identity Center will consider the windowsUpn value as the default value for "username". If you want to change the attribute mapping for "username" from your current mapping, confirm IAM Identity Center flows with dependency on "username" will continue to work as expected, before making the change.

If you use the [ListUsers](#) or [ListGroups](#) API actions or the [list-users](#) and [list-groups](#) AWS CLI commands to assign users and groups access to AWS accounts and to applications, you must specify the value for `AttributeValue` as an FQDN. This value must be in the following format: `user@example.com`. In the following example, `AttributeValue` is set to `janedoe@example.com`.

```
aws identitystore list-users --identity-store-id d-12345a678b --filters
  AttributePath=UserName,AttributeValue=janedoe@example.com
```

Supported IAM Identity Center attributes

The following table lists all IAM Identity Center attributes that are supported and that can be mapped to user attributes in your AWS Managed Microsoft AD directory. After you set up your application attribute mappings, you can use these same IAM Identity Center attributes to map to actual attributes used by that application.

Supported attributes in IAM Identity Center

`${user:AD_GUID}`

`${user:email}`

`${user:familyName}`

Supported attributes in IAM Identity Center

```
${user:givenName}
```

```
${user:middleName}
```

```
${user:name}
```

```
${user:preferredUsername}
```

```
${user:subject}
```

Supported external identity provider attributes

The following table lists all external identity provider (IdP) attributes that are supported and that can be mapped to attributes you can use when configuring [Attributes for access control](#) in IAM Identity Center. When using SAML assertions, you can use whichever attributes your IdP supports.

Supported attributes in your IdP

```
${path:userName}
```

```
${path:name.familyName}
```

```
${path:name.givenName}
```

```
${path:displayName}
```

```
${path:nickName}
```

```
${path:emails[primary eq true].value}
```

```
${path:addresses[type eq "work"].streetAddress}
```

```
${path:addresses[type eq "work"].locality}
```

```
${path:addresses[type eq "work"].region}
```

```
${path:addresses[type eq "work"].postalCode}
```

Supported attributes in your IdP

```
${path:addresses[type eq "work"].country}
```

```
${path:addresses[type eq "work"].formatted}
```

```
${path:phoneNumbers[type eq "work"].value}
```

```
${path:userType}
```

```
${path:title}
```

```
${path:locale}
```

```
${path:timezone}
```

```
${path:enterprise.employeeNumber}
```

```
${path:enterprise.costCenter}
```

```
${path:enterprise.organization}
```

```
${path:enterprise.division}
```

```
${path:enterprise.department}
```

```
${path:enterprise.manager.value}
```

Default mappings

The following table lists the default mappings for user attributes in IAM Identity Center to the user attributes in your AWS Managed Microsoft AD directory. IAM Identity Center only supports the list of attributes in the **User attribute in IAM Identity Center** column.

Note

If you don't have any assignments for your users and groups in IAM Identity Center when you enable configurable AD sync, the default mappings in the following table are used. For information about how to customize these mappings, see [Configure attribute mappings for your sync](#).

User attribute in IAM Identity Center	Maps to this attribute in your Microsoft AD directory
AD_GUID	<code>\${dir:guid}</code>
email *	<code>\${dir:windowsUpn}</code>
familyName	<code>\${dir:lastname}</code>
givenName	<code>\${dir:firstname}</code>
middleName	<code>\${dir:initials}</code>
name	<code>\${dir:displayname}</code>
preferredUsername	<code>\${dir:displayname}</code>
subject	<code>\${dir:windowsUpn}</code>

* The email attribute in IAM Identity Center must be unique within the directory. Otherwise, the JIT login process could fail.

You can change the default mappings or add more attributes to the SAML 2.0 assertion based on your requirements. For example, assume that your application requires the user's email in the `User.Email` SAML 2.0 attribute. In addition, assume that email addresses are stored in the `windowsUpn` attribute in your Microsoft AD directory. To achieve this mapping, you must make changes in the following two places in the IAM Identity Center console:

1. On the **Directory** page, under the **Attribute mappings** section, you would need to map the user attribute **email** to the `${dir:windowsUpn}` attribute (in the **Maps to this attribute in your directory** column)
2. On the **Applications** page, choose the application from the table. Choose the **Attribute mappings** tab. Then map the `User.Email` attribute to the `${user:email}` attribute (in the **Maps to this string value or user attribute in IAM Identity Center** column).

Note that you must supply each directory attribute in the form `${dir:AttributeName}`. For example, the `firstname` attribute in your Microsoft AD directory becomes `${dir:firstname}`. It

is important that every directory attribute have an actual value assigned. Attributes missing a value after `${dir:}` will cause user sign-in issues.

Map attributes in IAM Identity Center to attributes in your AWS Managed Microsoft AD directory

You can use the following procedure to specify how your user attributes in IAM Identity Center should map to corresponding attributes in your Microsoft AD directory.

To map attributes in IAM Identity Center to attributes in your directory

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Attributes for access control** tab, and then choose **Manage Attributes**.
4. On the **Manage attribute for access control** page, find the attribute in IAM Identity Center that you want to map and then type a value in the text box. For example, you might want to map the IAM Identity Center user attribute **email** to the Microsoft AD directory attribute `${dir:windowsUpn}`.
5. Choose **Save changes**.

Provision users and groups from Active Directory

IAM Identity Center provides the following two ways to provision users and groups from Active Directory.

- [IAM Identity Center configurable Active Directory \(AD\) sync \(recommended\)](#) — With this sync method, you can do the following:
 - Control data boundaries by explicitly defining the users and groups in Microsoft Active Directory that are automatically synchronized into IAM Identity Center. You can [add users and groups](#) or [remove users and groups](#) to change the scope of the sync at any time.
 - Assign synchronized users and groups single sign-on [access to AWS accounts](#) or [access to applications](#). The applications can be AWS managed applications or customer managed applications.
 - Control the synchronization process by [pausing and resuming the sync](#) as needed. This helps you regulate the load on production systems.

- [IAM Identity Center AD sync](#) — With this sync method, you use IAM Identity Center to assign users and groups in Active Directory access to AWS accounts and to applications. All identities with assignments are automatically synced into IAM Identity Center.

Topics

- [IAM Identity Center configurable AD sync](#)
- [IAM Identity Center AD sync](#)

IAM Identity Center configurable AD sync

IAM Identity Center configurable Active Directory (AD) sync enables you to explicitly configure the identities in Microsoft Active Directory that are automatically synchronized into IAM Identity Center and control the synchronization process.

The following topics provide information to enable you to configure and administer configurable AD sync.

Topics

- [Prerequisites and considerations](#)
- [How configurable AD sync works](#)
- [Configure and manage your sync scope](#)

Prerequisites and considerations

Before you use configurable AD sync, be aware of the following prerequisites and considerations:

- **Specifying users and groups in Active Directory to sync**

Before you can use IAM Identity Center to assign new users and groups access to AWS accounts and to AWS managed applications or customer managed applications, you must specify the users and groups in Active Directory to sync, and then sync them into IAM Identity Center.

- **AD sync** – When you make assignments for new users and groups by using the IAM Identity Center console or related assignment API actions, IAM Identity Center searches the domain controller directly for the specified users or groups, completes the assignment, and then periodically syncs the user or group metadata into IAM Identity Center.

- **Configurable AD sync** – IAM Identity Center doesn't search your domain controller directly for users and groups. Instead, you must first specify the list of users and groups to sync. You can configure this list, also known as the *sync scope*, in one of the following ways, depending on whether you have users and groups that are already synced into IAM Identity Center, or you have new users and groups that you are syncing for the first time by using configurable AD sync.
 - Existing users and groups: If you have users and groups that are already synced into IAM Identity Center, the sync scope in configurable AD sync is prepopulated with a list of those users and groups. To assign new users or groups, you must specifically add them to the sync scope. For more information, see [Add users and groups to your sync scope](#).
 - New users and groups: If you want to assign new users and groups access to AWS accounts and to applications, you must specify which users and groups to add to the sync scope in configurable AD sync before you can use IAM Identity Center to make the assignment. For more information, see [Add users and groups to your sync scope](#).

- **Making assignments to nested groups in Active Directory**

Groups that are members of other groups are called *nested groups* (or child groups). When you make assignments to a group in Active Directory that contains nested groups, the way in which the assignments are applied depends on whether you use AD sync or configurable AD sync.

- **AD sync** – When you make assignments to a group in Active Directory that contains nested groups, only the direct members of the group can access the account. For example, if you assign access to Group A, and Group B is a member of Group A, only the direct members of Group A can access the account. No members of Group B inherit the access.
 - **Configurable AD sync** – Using configurable AD sync to make assignments to a group in Active Directory that contains nested groups might increase the scope of users who have access to AWS accounts or to applications. In this case, the assignment applies to all users, including those in nested groups. For example, if you assign access to Group A, and Group B is a member of Group A, members of Group B also inherit this access.
- **Updating automated workflows**

If you have automated workflows that use the IAM Identity Center identity store API actions and IAM Identity Center assignment API actions to assign new users and groups access to accounts and to applications, and to sync them into IAM Identity Center, you must adjust those workflows by April 15, 2022 so that they function as expected with configurable AD sync. Configurable AD

sync changes the order in which user and group assignment and provisioning occur, and the way in which queries are performed.

- **AD sync** – The process of assignments occurs first. You assign users and groups access to AWS accounts and to applications. After the users and groups are assigned access, they are automatically provisioned (synced into IAM Identity Center). If you have an automated workflow, this means that when you add a new user to Active Directory, your automated workflow can query Active Directory for the user by using the identity store `ListUser` API action, and then assign the user access by using the IAM Identity Center assignment API actions. Because the user has an assignment, that user is automatically provisioned into IAM Identity Center.
- **Configurable AD sync** – Provisioning occurs first, and it is not automatically performed. Instead, you must first explicitly add users and groups to the identity store by adding them to your sync scope. For information about the recommended steps for automating your sync configuration for configurable AD sync, see [Automate your sync configuration for configurable AD sync](#).

How configurable AD sync works

IAM Identity Center refreshes the AD-based identity data in the identity store by using the following process.

Creation

After you connect your self-managed directory in Active Directory or your AWS Managed Microsoft AD directory that is managed by AWS Directory Service to IAM Identity Center, you can explicitly configure the Active Directory users and groups that you want to sync into the IAM Identity Center identity store. The identities that you choose will be synchronized every three hours or so into the IAM Identity Center identity store. Depending on the size of your directory, the sync process might take longer.

Groups that are members of other groups (called *nested groups* or *child groups*) are also written to the identity store. When you make assignments to a group in Active Directory that contains nested groups, the way in which the assignments are applied depends on whether you use AD sync or configurable AD sync. For more information, see [Making assignments to nested groups in Active Directory](#).

You can only assign access to new users or groups after they are synchronized into the IAM Identity Center identity store.

Update

The identity data in the IAM Identity Center identity store stays fresh by periodically reading data from the source directory in Active Directory. IAM Identity Center syncs data from your Active Directory every hour in a sync cycle by default. It may take 30 minutes to 2 hours for the data to sync into IAM Identity Center, based on the size of your Active Directory.

User and group objects that are in the sync scope and their memberships are created or updated in IAM Identity Center to map to the corresponding objects in the source directory in Active Directory. For user attributes, only the subset of attributes listed in the **Attributes for access control** section of the IAM Identity Center console are updated in IAM Identity Center. It may take one sync cycle for any attribute updates you make in Active Directory to reflect in IAM Identity Center.

You can also update the subset of users and groups that you synchronize into the IAM Identity Center identity store. You can choose to add new users or groups to this subset, or remove them. Any identities that you add are synchronized at the next scheduled sync. Identities that you remove from the subset will stop being updated in the IAM Identity Center identity store. Any user who isn't synchronized for more than 28 days will be disabled in the IAM Identity Center identity store. The corresponding user objects will be automatically disabled in the IAM Identity Center identity store during the next sync cycle, unless they are part of another group that is still part of the sync scope.

Deletion

Users and groups are deleted from the IAM Identity Center identity store when the corresponding user or group objects are deleted from the source directory in Active Directory. Alternatively, you can explicitly delete user objects from the IAM Identity Center identity store by using the IAM Identity Center console. If you use the IAM Identity Center console, you must also remove the users from the sync scope to ensure that they aren't re-synced back into IAM Identity Center during the next sync cycle.

You can also pause and restart synchronization at any time. If you pause synchronization for more than 28 days, all your users will be disabled.

Configure and manage your sync scope

You can configure your sync scope in either of the following ways:

- **Guided setup:** If you are synchronizing your users and groups from Active Directory into IAM Identity Center for the first time, follow the steps in [Guided setup](#) to configure your sync scope.

After you complete the guided setup, you can modify your sync scope at any time by following the other procedures in this section.

- If you already have users and groups that are synchronized into IAM Identity Center or you don't want to follow the guided setup, choose **Manage sync**. Skip the guided setup procedure and follow the other procedures in this section as required to configure and manage your sync scope.

Procedures

- [Guided setup](#)
- [Add users and groups to your sync scope](#)
- [Remove users and groups from your sync scope](#)
- [Pause and resume your sync](#)
- [Configure attribute mappings for your sync](#)
- [Automate your sync configuration for configurable AD sync](#)

Guided setup

1. Open the [IAM Identity Center console](#).

Note

Make sure that the IAM Identity Center console is using one of the AWS Regions where your AWS Managed Microsoft AD directory is located before you move to the next step.

2. Choose **Settings**.
3. At the top of the page, in the notification message, choose **Start guided setup**.
4. In **Step 1 – optional: Configure attribute mappings**, review the default user and group attribute mappings. If no changes are required, choose **Next**. If changes are required, make the changes, and then choose **Save changes**.
5. In **Step 2 – optional: Configure sync scope**, choose the **Users** tab. Then, enter the exact username of the user that you want to add to your sync scope and choose **Add**. Next, choose the **Groups** tab. Enter the exact group name of the group that you want to add to your sync scope and choose **Add**. Then, choose **Next**. If you want to add users and groups to your sync scope later, make no changes and choose **Next**.

6. In **Step 3: Review and save configuration**, confirm your **Attribute mappings** in **Step 1: Attribute mappings** and your **Users and groups** in **Step 2: Sync scope**. Choose **Save configuration**. This takes you to the **Manage Sync** page.

Add users and groups to your sync scope

To add users

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. On the **Manage Sync** page, choose the **Users** tab, and then choose **Add users and groups**.
5. On the **Users** tab, under **User**, enter the exact user name and choose **Add**.
6. Under **Added Users and Groups**, review the user that you want to add.
7. Choose **Submit**.
8. In the navigation pane, choose **Users**.
9. On the **Users** page, it might take some time for the user that you specified to appear in the list. Choose the refresh icon to update the list of users.

To add groups

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. On the **Manage Sync** page, choose the **Groups** tab, and then choose **Add users and groups**.
5. Choose the **Groups** tab. Under **Group**, enter the exact group name and choose **Add**.
6. Under **Added Users and Groups**, review the group that you want to add.
7. Choose **Submit**.
8. In the navigation pane, choose **Groups**.
9. On the **Groups** page, it might take some time for the group that you specified to appear in the list. Choose the refresh icon to update the list of groups.

Remove users and groups from your sync scope

For more information about what happens when you remove users and groups from your sync scope, see [How configurable AD sync works](#).

To remove users

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. Choose the **Users** tab.
5. Under **Users in sync scope**, select the check box beside the user that you want to delete. To delete all users, select the check box beside **Username**.
6. Choose **Remove**.

To remove groups

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. Choose the **Groups** tab.
5. Under **Groups in sync scope**, select the check box beside the user that you want to delete. To delete all groups, select the check box beside **Group name**.
6. Choose **Remove**.

Pause and resume your sync

Pausing your sync pauses all future sync cycles and prevents any changes that you make to users and groups in Active Directory from being reflected in IAM Identity Center. After you resume the sync, the sync cycle picks up these changes from the next scheduled sync.

To pause your sync

1. Open the [IAM Identity Center console](#).

2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. Under **Manage Sync**, choose **Pause sync**.

To resume your sync

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. Under **Manage Sync**, choose **Resume sync**.

Note

If you see **Pause sync** instead of **Resume sync**, the sync from Active Directory to IAM Identity Center has already resumed.

Configure attribute mappings for your sync

For more information about available attributes, see [Attribute mappings for AWS Managed Microsoft AD directory](#).

To configure attribute mappings in IAM Identity Center to your directory

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage Sync**.
4. Under **Manage Sync**, choose **View attribute mapping**.
5. Under **Active Directory user attributes**, configure **IAM Identity Center identity store attributes** and **Active Directory user attributes**. For example, you might want to map the IAM Identity Center identity store attribute `email` to the Active Directory user directory attribute `${objectguid}`.

Note

Under **Group attributes**, **IAM Identity Center identity store attributes** and **Active Directory group attributes** can't be changed.

6. Choose **Save changes**. This returns you to the **Manage Sync** page.

Automate your sync configuration for configurable AD sync

To ensure that your automated workflow works as expected with configurable AD sync, we recommend that you perform the following steps to automate your sync configuration.

To automate your sync configuration for configurable AD sync

1. In Active Directory, create a *parent sync group* to contain all users and groups that you want to sync into IAM Identity Center. For example, you can name the group *IAMIdentityCenterAllUsersAndGroups*.
2. In IAM Identity Center, add the parent sync group to your configurable sync list. IAM Identity Center will synchronize all users, groups, sub-groups, and members of all groups contained within the parent sync group.
3. Use the Active Directory user and group management API actions provided by Microsoft to add or remove users and groups from the parent sync group.

IAM Identity Center AD sync

With IAM Identity Center AD sync, you use IAM Identity Center to assign users and groups in Active Directory access to AWS accounts and to AWS managed applications or customer managed applications. All identities with assignments are automatically synced into IAM Identity Center.

How IAM Identity Center AD sync works

IAM Identity Center refreshes the AD-based identity data in the identity store using the following process.

Creation

When you assign users or groups to AWS accounts or applications by using the AWS console or the assignment API calls, information about the users, groups, and membership is periodically

synchronized into the IAM Identity Center identity store. Users or groups that are added to IAM Identity Center assignments usually appear in the AWS identity store within two hours. Depending on the amount of data being synchronized, this process might take longer. Only users and groups that are directly assigned access, or are members of a group that is assigned access, are synchronized.

Groups that are members of other groups (called nested groups) are also written to the identity store. When you make assignments to a group in Active Directory that contains nested groups, the way in which the assignments are applied depends on whether you use AD sync or configurable AD sync.

- **AD sync** – When you make assignments to a group in Active Directory that contains nested groups, only the direct members of the group can access the account. For example, if you assign access to Group A, and Group B is a member of Group A, only the direct members of Group A can access the account. No members of Group B inherit the access.
- **Configurable AD sync** – Using configurable AD sync to make assignments to a group in Active Directory that contains nested groups might increase the scope of users who have access to AWS accounts or to applications. In this case, the assignment applies to all users, including those in nested groups. For example, if you assign access to Group A, and Group B is a member of Group A, members of Group B also inherit this access.

If a user accesses IAM Identity Center before their user object has been synchronized for the first time, that user's identity store object is created on demand using just-in-time (JIT) provisioning. Users created by JIT provisioning are not synchronized unless they have directly assigned or group-based IAM Identity Center entitlements. Group memberships for JIT-provisioned users are unavailable until after synchronization.

For instructions on how to assign users access to AWS accounts, see [Single sign-on access to AWS accounts](#).

Update

The identity data in the IAM Identity Center identity store stays fresh by periodically reading data from the source directory in Active Directory. Identity data that is changed in Active Directory will usually appear in the AWS identity store within four hours. Depending on the amount of data being synchronized, this process might take longer.

User and group objects and their memberships are created or updated in IAM Identity Center to map to the corresponding objects in the source directory in Active Directory. For user attributes,

only the subset of attributes listed in the **Manage attributes for access control** section of the IAM Identity Center console is updated in IAM Identity Center. In addition, user attributes are updated with each user authentication event.

Deletion

Users and groups are deleted from the IAM Identity Center identity store when the corresponding user or group objects are deleted from the source directory in Active Directory.

Manage an external identity provider

With IAM Identity Center, you can connect your existing workforce identities from external identity providers (IdPs) through the Security Assertion Markup Language (SAML) 2.0 and System for Cross-Domain Identity Management (SCIM) protocols. This enables your users to sign in to the AWS access portal with their corporate credentials. They can then navigate to their assigned accounts, roles, and applications hosted in external IdPs.

For example, you can connect an external IdP such as Okta or Microsoft Entra ID, to IAM Identity Center. Your users can then sign in to the AWS access portal with their existing Okta or Microsoft Entra ID credentials. To control what your users can do once they've signed in, you can assign them access permissions centrally across all the accounts and applications in your AWS organization. In addition, developers can simply sign in to the AWS Command Line Interface (AWS CLI) using their existing credentials, and benefit from automatic short-term credential generation and rotation.

If you're using a self-managed directory in Active Directory or an AWS Managed Microsoft AD, see [Connect to a Microsoft AD directory](#).

Note

The SAML protocol does not provide a way to query the IdP to learn about users and groups. Therefore, you must make IAM Identity Center aware of those users and groups by provisioning them into IAM Identity Center.

Topics

- [Provisioning when users come from an external IdP](#)
- [How to connect to an external identity provider](#)
- [How to change an external identity provider's metadata in IAM Identity Center](#)

- [Using SAML and SCIM identity federation with external identity providers](#)
- [SCIM profile and SAML 2.0 implementation](#)

Provisioning when users come from an external IdP

When using an external IdP, you must provision all applicable users and groups into IAM Identity Center before you can make any assignments to AWS accounts or applications. To do this, you can configure [Provisioning an external identity provider into IAM Identity Center using SCIM](#) for your users and groups, or use [Manual provisioning](#). Regardless of how you provision users, IAM Identity Center redirects the AWS Management Console, command line interface, and application authentication to your external IdP. IAM Identity Center then grants access to those resources based on policies you create in IAM Identity Center. For more information about provisioning, see [User and group provisioning](#).

How to connect to an external identity provider

There are different prerequisites, considerations, and provisioning procedures for the supported external IdPs. There are step-by-step tutorials available for several IdPs:

- [CyberArk](#)
- [Google Workspace](#)
- [JumpCloud](#)
- [Microsoft Entra ID](#)
- [Okta](#)
- [OneLogin](#)
- [Ping Identity](#)

For more information on the considerations for external IdPs that IAM Identity Center supports, see [Using SAML and SCIM identity federation with external identity providers](#).

The following procedure provides a general overview of the procedure that's used with all external identity providers.

To connect to an external identity provider

1. Open the [IAM Identity Center console](#).

2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Change identity source**.
4. Under **Choose identity source**, select **External identity provider**, and then choose **Next**.
5. Under **Configure external identity provider**, do the following:
 - a. Under **Service provider metadata**, choose **Download metadata file** to download the metadata file and save it on your system. The IAM Identity Center SAML metadata file is required by your external identity provider.
 - b. Under **Identity provider metadata**, choose **Choose file**, and locate the metadata file that you downloaded from your external identity provider. Then upload the file. This metadata file contains the necessary public x509 certificate used to trust messages that are sent from the IdP.
 - c. Choose **Next**.

 **Important**

Changing your source to or from Active Directory removes all existing user and group assignments. You must manually reapply assignments after you have successfully changed your source.

6. After you read the disclaimer and are ready to proceed, enter **ACCEPT**.
7. Choose **Change identity source**. A status message informs you that you successfully changed the identity source.

How to change an external identity provider's metadata in IAM Identity Center

You can change your external identity provider's metadata which you previously supplied to the IAM Identity Center. These changes affect your users' ability to sign in and access AWS resources through IAM Identity Center. The following procedure describes how to update your external IdP's metadata that's stored in IAM Identity Center.

To change an external identity provider's metadata

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.

3. On the **Settings** page, choose the **Identity source** tab. Choose **Actions** and then choose **Manage Authentication**.
4. In the **Identity provider metadata** section, choose **Edit IdP metadata**. You can make the changes to the IdP sign-in URL and or IdP issuer URL for your external IdP on this page. Choose **Save changes** when you've made all the necessary changes.

Using SAML and SCIM identity federation with external identity providers

IAM Identity Center implements the following standards-based protocols for identity federation:

- SAML 2.0 for user authentication
- SCIM for provisioning

Any identity provider (IdP) that implements these standard protocols is expected to interoperate successfully with IAM Identity Center, with the following special considerations:

• SAML

- IAM Identity Center requires a SAML nameID format of email address (that is, `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`).
- The value of the nameID field in assertions must be an RFC 2822 (<https://tools.ietf.org/html/rfc2822>) addr-spec compliant ("name@domain.com") string (<https://tools.ietf.org/html/rfc2822#section-3.4.1>).
- The metadata file cannot be over 75000 characters.
- The metadata must contain an entityId, X509 certificate, and SingleSignOnService as part of the sign-in URL.
- An encryption key is not supported.

• SCIM

- The IAM Identity Center SCIM implementation is based on SCIM RFCs 7642 (<https://tools.ietf.org/html/rfc7642>), 7643 (<https://tools.ietf.org/html/rfc7643>), and 7644 (<https://tools.ietf.org/html/rfc7644>), and the interoperability requirements laid out in the March 2020 draft of the FastFed Basic SCIM Profile 1.0 (https://openid.net/specs/fastfed-scim-1_0-02.html#rfc.section.4). Any differences between these documents and the current implementation in IAM Identity Center are described in the [Supported API operations](#) section of the *IAM Identity Center SCIM Implementation Developer Guide*.

IdPs that do not conform to the standards and considerations mentioned above are not supported. Please contact your IdP for questions or clarifications regarding the conformance of their products to these standards and considerations.

If you have any issues connecting your IdP to IAM Identity Center, we recommend that you check:

- AWS CloudTrail logs by filtering on the event name **ExternalIdPDirectoryLogin**
- IdP-specific logs and/or debug logs
- [Troubleshooting IAM Identity Center issues](#)

Note

Some IdPs, such as the ones in the [Getting started tutorials for IAM Identity Center](#), offer a simplified configuration experience for IAM Identity Center in the form of an “application” or “connector” built specifically for IAM Identity Center. If your IdP provides this option, we recommend that you use it, being careful to choose the item that’s built specifically for IAM Identity Center. Other items called “AWS”, “AWS federation”, or similar generic “AWS” names may use other federation approaches and/or endpoints, and may not work as expected with IAM Identity Center.

SCIM profile and SAML 2.0 implementation

Both SCIM and SAML are important considerations for configuring IAM Identity Center.

SAML 2.0 implementation

IAM Identity Center supports identity federation with [SAML \(Security Assertion Markup Language\) 2.0](#). This allows IAM Identity Center to authenticate identities from external identity providers (IdPs). SAML 2.0 is an open standard used for securely exchanging SAML assertions. SAML 2.0 passes information about a user between a SAML authority (called an identity provider or IdP), and a SAML consumer (called a service provider or SP). The IAM Identity Center service uses this information to provide federated single sign-on. Single sign-on allows users to access AWS accounts and configured applications based on their existing identity provider credentials.

IAM Identity Center adds SAML IdP capabilities to your IAM Identity Center store, AWS Managed Microsoft AD, or to an external identity provider. Users can then single sign-on into services that

support SAML, including the AWS Management Console and third-party applications such as Microsoft 365, Concur, and Salesforce.

The SAML protocol however doesn't provide a way to query the IdP to learn about users and groups. Therefore, you must make IAM Identity Center aware of those users and groups by provisioning them into IAM Identity Center.

SCIM profile

IAM Identity Center provides support for the System for Cross-domain Identity Management (SCIM) v2.0 standard. SCIM keeps your IAM Identity Center identities in sync with identities from your IdP. This includes any provisioning, updates, and deprovisioning of users between your IdP and IAM Identity Center.

For more information about how to implement SCIM, see [Provisioning an external identity provider into IAM Identity Center using SCIM](#). For additional details about IAM Identity Center's SCIM implementation, see the [IAM Identity Center SCIM Implementation Developer Guide](#).

Topics

- [Provisioning an external identity provider into IAM Identity Center using SCIM](#)
- [Manage SAML 2.0 certificates](#)

Provisioning an external identity provider into IAM Identity Center using SCIM

IAM Identity Center supports automatic provisioning (synchronization) of user and group information from your identity provider (IdP) into IAM Identity Center using the System for Cross-domain Identity Management (SCIM) v2.0 protocol. When you configure SCIM synchronization, you create a mapping of your identity provider (IdP) user attributes to the named attributes in IAM Identity Center. This causes the expected attributes to match between IAM Identity Center and your IdP. You configure this connection in your IdP using your SCIM endpoint for IAM Identity Center and a bearer token that you create in IAM Identity Center.

Topics

- [Considerations for using automatic provisioning](#)
- [How to monitor access token expiry](#)
- [How to enable automatic provisioning](#)
- [How to disable automatic provisioning](#)

- [How to generate a new access token](#)
- [How to delete an access token](#)
- [How to rotate an access token](#)
- [Manual provisioning](#)

Considerations for using automatic provisioning

Before you begin deploying SCIM, we recommend that you first review the following important considerations about how it works with IAM Identity Center. For additional provisioning considerations, see the [Getting started tutorials for IAM Identity Center](#) applicable to your IdP.

- If you are provisioning a primary email address, this attribute value must be unique for each user. In some IdPs, the primary email address might not be a real email address. For example, it might be a Universal Principal Name (UPN) that only looks like an email. These IdPs may have a secondary or “other” email address that contains the user’s real email address. You must configure SCIM in your IdP to map the non-Null unique email address to the IAM Identity Center primary email address attribute. And you must map the users non-Null unique sign-in identifier to the IAM Identity Center user name attribute. Check to see whether your IdP has a single value that is both the sign-in identifier and the user’s email name. If so, you can map that IdP field to both the IAM Identity Center primary email and the IAM Identity Center user name.
- For SCIM synchronization to work, every user must have a **First name**, **Last name**, **Username** and **Display name** value specified. If any of these values are missing from a user, that user will not be provisioned.
- If you need to use third-party applications, you will first need to map the outbound SAML subject attribute to the user name attribute. If the third-party application needs a routable email address, you must provide the email attribute to your IdP.
- SCIM provisioning and update intervals are controlled by your identity provider. Changes to users and groups in your identity provider are only reflected in IAM Identity Center after your identity provider sends those changes to IAM Identity Center. Check with your identity provider for details on the frequency of user and group updates.
- Currently, multivalued attributes (such as multiple emails or phone numbers for a given user) are not provisioned with SCIM. Attempts to synchronize multivalued attributes into IAM Identity Center with SCIM will fail. To avoid failures, ensure that only a single value is passed for each attribute. If you have users with multivalued attributes, remove or modify the duplicate attribute mappings in SCIM at your IdP for the connection to IAM Identity Center.

- Verify that the `externalId` SCIM mapping at your IdP corresponds to a value that is unique, always present, and least likely to change for your users. For example, your IdP might provide a guaranteed `objectId` or other identifier that's not affected by changes to user attributes like name and email. If so, you can map that value to the SCIM `externalId` field. This ensures that your users won't lose AWS entitlements, assignments, or permissions if you need to change their name or email.
- Users who have not yet been assigned to an application or AWS account cannot be provisioned into IAM Identity Center. To synchronize users and groups, make sure that they are assigned to the application or other setup that represents your IdP's connection to IAM Identity Center.
- User deprovisioning behavior is managed by the identity provider and may vary by their implementation. Check with your identity provider for details on user deprovisioning.

For more information about IAM Identity Center's SCIM implementation, see the [IAM Identity Center SCIM Implementation Developer Guide](#).

How to monitor access token expiry

SCIM access tokens are generated with a validity of one year. When your SCIM access token is set to expire in 90 days or less, AWS sends you reminders in the IAM Identity Center console and over the AWS Health Dashboard to help you rotate the token. By rotating the SCIM access token before it expires, you continually secure automatic provisioning of user and group information. If the SCIM access token expires, the synchronization of user and group information from your identity provider into IAM Identity Center stops, so automatic provisioning can no longer make updates or create and delete information. Disruption to automatic provisioning may impose increased security risks and impact access to your services.

The Identity Center console reminders persist until you rotate the SCIM access token and delete any unused or expired access tokens. The AWS Health Dashboard events are renewed weekly between 90 to 60 days, twice per week from 60 to 30 days, three times per week from 30 to 15 days, and daily from 15 days until the SCIM access tokens expires.

How to enable automatic provisioning

Use the following procedure to enable automatic provisioning of users and groups from your IdP to IAM Identity Center using the SCIM protocol.

Note

Before you begin this procedure, we recommend that you first review provisioning considerations that are applicable to your IdP. For more information, see the [Getting started tutorials for IAM Identity Center](#) for your IdP.

To enable automatic provisioning in IAM Identity Center

1. After you have completed the prerequisites, open the [IAM Identity Center console](#).
2. Choose **Settings** in the left navigation pane.
3. On the **Settings** page, locate the **Automatic provisioning** information box, and then choose **Enable**. This immediately enables automatic provisioning in IAM Identity Center and displays the necessary SCIM endpoint and access token information.
4. In the **Inbound automatic provisioning** dialog box, copy each of the values for the following options. You will need to paste these in later when you configure provisioning in your IdP.
 - a. **SCIM endpoint** - For example, `https://scim.us-east-2.amazonaws.com/1111111111-2222-3333-4444-555555555555/scim/v2`
 - b. **Access token** - Choose **Show token** to copy the value.

Warning

This is the only time where you can obtain the SCIM endpoint and access token. Ensure you copy these values before moving forward. You will enter these values to configure automatic provisioning in Okta later in this tutorial.

5. Choose **Close**.

After you complete this procedure, you must configure automatic provisioning in your IdP. For more information, see the [Getting started tutorials for IAM Identity Center](#) for your IdP.

How to disable automatic provisioning

Use the following procedure to disable automatic provisioning in the IAM Identity Center console.

⚠ Important

You must delete the access token before you start this procedure. For more information, see [How to delete an access token](#).

To disable automatic provisioning in the IAM Identity Center console

1. In the [IAM Identity Center console](#), choose **Settings** in the left navigation pane.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage provisioning**.
3. On the **Automatic provisioning** page, choose **Disable**.
4. In the **Disable automatic provisioning** dialog box, review the information, type **DISABLE**, and then choose **Disable automatic provisioning**.

How to generate a new access token

Use the following procedure to generate a new access token in the IAM Identity Center console.

ℹ Note

This procedure requires that you have previously enabled automatic provisioning. For more information, see [How to enable automatic provisioning](#).

To generate a new access token

1. In the [IAM Identity Center console](#), choose **Settings** in the left navigation pane.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage provisioning**.
3. On the **Automatic provisioning** page, under **Access tokens**, choose **Generate token**.
4. In the **Generate new access token** dialog box, copy the new access token and save it in a safe place.
5. Choose **Close**.

How to delete an access token

Use the following procedure to delete an existing access token in the IAM Identity Center console.

To delete an existing access token

1. In the [IAM Identity Center console](#), choose **Settings** in the left navigation pane.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage provisioning**.
3. On the **Automatic provisioning** page, under **Access tokens**, select the access token you want to delete, and then choose **Delete**.
4. In the **Delete access token** dialog box, review the information, type **DELETE**, and then choose **Delete access token**.

How to rotate an access token

An IAM Identity Center directory supports up to two access tokens at a time. To generate an additional access token prior to any rotation, delete any expired or unused access tokens.

If your SCIM access token is close to expiring, you can use the following procedure to rotate an existing access token in the IAM Identity Center console.

To rotate an access token

1. In the [IAM Identity Center console](#), choose **Settings** in the left navigation pane.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage provisioning**.
3. On the **Automatic provisioning** page, under **Access tokens**, make a note of the token ID of the token you want to rotate.
4. Follow the steps in [How to generate a new access token](#) to create a new token. If you have already created the maximum number of SCIM access tokens, you will first need to delete one of the existing tokens.
5. Go to your identity provider's website and configure the new access token for SCIM provisioning, and then test connectivity to IAM Identity Center using the new SCIM access token. Once you've confirmed that provisioning is working successfully using the new token, continue to the next step in this procedure.

6. Follow the steps in [How to delete an access token](#) to delete the old access token you noted earlier. You can also use the token's creation date as a hint for which token to remove.

Manual provisioning

Some IdPs don't have System for Cross-domain Identity Management (SCIM) support or have an incompatible SCIM implementation. In those cases, you can manually provision users through the IAM Identity Center console. When you add users to IAM Identity Center, ensure that you set the user name to be identical to the user name that you have in your IdP. At a minimum, you must have a unique email address and user name. For more information, see [User name and email address uniqueness](#).

You must also manage all groups manually in IAM Identity Center. To do this, you create the groups and add them using the IAM Identity Center console. These groups do not need to match what exists in your IdP. For more information, see [Groups](#).

Manage SAML 2.0 certificates

IAM Identity Center uses certificates to set up a SAML trust relationship between IAM Identity Center and your external identity provider (IdP). When you add an external IdP in IAM Identity Center, you must also obtain at least one public SAML 2.0 X.509 certificate from the external IdP. That certificate is usually installed automatically during the IdP SAML metadata exchange during trust creation.

As an IAM Identity Center administrator, you'll occasionally need to replace older IdP certificates with newer ones. For example, you might need to replace an IdP certificate when the expiration date on the certificate approaches. The process of replacing an older certificate with a newer one is referred to as certificate rotation.


Topics

- [Rotate a SAML 2.0 certificate](#)
- [Certificate expiration status indicators](#)

Rotate a SAML 2.0 certificate

You may need to import certificates periodically in order to rotate invalid or expired certificates issued by your identity provider. This helps to prevent authentication disruption or downtime. All imported certificates are automatically active. Certificates should only be deleted after ensuring that they are no longer in use with the associated identity provider.

You should also consider that some IdPs might not support multiple certificates. In this case, the act of rotating certificates with these IdPs might mean a temporary service disruption for your users. Service is restored when the trust with that IdP has been successfully reestablished. Plan this operation carefully during off peak hours if possible.

 **Note**

As a security best practice, upon any signs of compromise or mishandling of an existing SAML certificate, you should immediately remove and rotate the certificate.

Rotating an IAM Identity Center certificate is a multistep process that involves the following:

- Obtaining a new certificate from the IdP
- Importing the new certificate into IAM Identity Center
- Activating the new certificate in the IdP
- Deleting the older certificate

Use all of the following procedures to complete the certificate rotation process while avoiding any authentication downtime.

Step 1: Obtain a new certificate from the IdP

Go to the IdP website and download their SAML 2.0 certificate. Make sure that the certificate file is downloaded in PEM encoded format. Most providers allow you to create multiple SAML 2.0 certificates in the IdP. It is likely that these will be marked as disabled or inactive.

Step 2: Import the new certificate into IAM Identity Center

Use the following procedure to import the new certificate using the IAM Identity Center console.

1. In the [IAM Identity Center console](#), choose **Settings**.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage authentication**.
3. On the **Manage SAML 2.0 certificates** page, choose **Import certificate**.
4. On the **Import SAML 2.0 certificate** dialog, choose **Choose file**, navigate to your certificate file and select it, and then choose **Import certificate**.

At this point, IAM Identity Center will trust all incoming SAML messages signed from both of the certificates that you have imported.

Step 3: Activate the new certificate in the IdP

Go back to the IdP website and mark the new certificate that you created earlier as primary or active. At this point all SAML messages signed by the IdP should be using the new certificate.

Step 4: Delete the old certificate

Use the following procedure to complete the certificate rotation process for your IdP. There must always be at least one valid certificate listed, and it cannot be removed.

Note

Make sure that your identity provider is no longer signing SAML responses with this certificate before deleting it.

1. On the **Manage SAML 2.0 certificates** page, choose the certificate that you want to delete. Choose **Delete**.
2. In the **Delete SAML 2.0 certificate** dialog box, type **DELETE** to confirm, and then choose **Delete**.
3. Return to the IdP's website and perform the necessary steps to remove the older inactive certificate.

Certificate expiration status indicators

While on the **Manage SAML 2.0 certificates** page, you might notice colored status indicator icons. These icons appear in the **Expires on** column next to each certificate in the list. The following describes the criteria that IAM Identity Center uses to determine which icon is displayed for each certificate.

- **Red** – Indicates that a certificate is currently expired.
- **Yellow** – Indicates that a certificate will expire in 90 days or less.
- **Green** – Indicates that a certificate is currently valid and will remain valid for at least 90 more days.

To check the current status of a certificate

1. In the [IAM Identity Center console](#), choose **Settings**.
2. On the **Settings** page, choose the **Identity source** tab, and then choose **Actions > Manage authentication**.
3. On the **Manage SAML 2.0 authentication** page, under **Manage SAML 2.0 certificates**, review the status of the certificates in the list as indicated in the **Expires on** column.

Using the AWS access portal

The AWS access portal provides you (end users) with single sign-on access to all your AWS accounts and most commonly used cloud applications such as Office 365, Concur, Salesforce, and many more. You can quickly launch multiple applications simply by choosing the AWS account or application icon in the portal. The presence of application icons in your AWS access portal means that an administrator from your company has granted you access to those AWS accounts or applications. It also means that you can access all these accounts or applications from the AWS access portal without additional sign-in prompts.

Contact your administrator to request additional access in the following situations:

- You don't see an AWS account or application that you need to access.
- The access that you have to a given account or application isn't what you expected.

Topics

- [Activating the AWS access portal for first-time IAM Identity Center users](#)
- [Signing in to the AWS access portal](#)
- [Resetting your AWS access portal user password](#)
- [Getting IAM Identity Center user credentials for the AWS CLI or AWS SDKs](#)
- [Creating shortcut links to AWS Management Console destinations](#)
- [Registering your device for MFA](#)
- [Customizing the AWS access portal URL](#)

Activating the AWS access portal for first-time IAM Identity Center users

If this is your first time attempting to sign in to the AWS access portal, check your email for instructions on how to activate your user credentials.

To activate your user credentials

1. Depending on the email you received from your company, choose one of the following methods to activate your user credentials so that you can start using the AWS access portal.
 - a. If you received an email with the subject **Invitation to join AWS IAM Identity Center (successor to AWS Single Sign-On)**, open it and choose **Accept invitation**. On the **New user sign up** page, enter and confirm a password, and then choose **Set new password**. You'll use that password each time you sign in to the portal.
 - b. If you were sent an email from your company's IT support or IT administrator, follow the instructions they provided to activate your user credentials.
2. After you activate your user credentials by providing a new password, the AWS access portal signs you in automatically. If this doesn't occur, you can manually sign in to the AWS access portal by using the instructions provided in the next section.

Signing in to the AWS access portal

By this time, you should have been provided a specific sign-in URL to the AWS access portal by an administrator. Once you have this URL, you can proceed with signing in to the portal. For more information, see [Sign in to the AWS access portal](#).

Note

After you sign in, the default duration for your AWS access portal session is 8 hours. Be aware that an administrator can [change the duration](#) of this session.

Trusted devices

When you choose the option **This is a trusted device** from the sign-in page, IAM Identity Center considers all future sign-ins from that device as authorized. This means that IAM Identity Center

won't present an option to enter in an MFA code as long as you're using that trusted device. However, there are some exceptions, including signing in from a new browser or when your device has been issued an unknown IP address.

Sign in tips for the AWS access portal

Here are some tips to help you manage your AWS access portal experience.

- Occasionally, you might need to sign out and sign back in to the AWS access portal. This might be necessary to access new applications that your administrator recently assigned to you. This is not required, however, because all new applications are refreshed every hour.
- When you sign in to the AWS access portal, you can open any of the applications listed in the portal by choosing the application's icon. After you are done using the application, you can either close the application or sign out of the AWS access portal. Closing the application signs you out of that application only. Any other applications that you have opened from the AWS access portal remain open and running.
- Before you can sign in as a different user, you must first sign out of the AWS access portal. Signing out from the portal completely removes your credentials from the browser session.
- Once you sign in to the AWS access portal, you can switch to a role. Switching roles temporarily sets aside your original user permissions and instead gives you the permissions assigned to the role. For more information, see [Switching to a role \(console\)](#).

Signing out of the AWS access portal

When you sign out from the portal, your credentials are completely removed from the browser session. For more information, see [Sign out of the AWS access portal](#) in the *AWS Sign-In* guide.

To sign out of the AWS access portal

- In the AWS access portal, choose **Sign out** from the navigation bar.

Note

If you want to sign in as a different user, you must first sign out of the AWS access portal.

Resetting your AWS access portal user password

The AWS access portal provides [IAM Identity Center](#) users with single sign-on access to all their assigned AWS accounts and cloud applications through a web portal. The AWS access portal is different from the [AWS Management Console](#), which is a collection of service consoles for managing AWS resources.

Use this procedure to reset your IAM Identity Center user password for the AWS access portal. Learn more about [User types](#) in the *AWS Sign-In User Guide*.

Considerations

The reset your password functionality for your AWS access portal is only available for users of Identity Center instances that are using Identity Center directory or [AWS Managed Microsoft AD](#) as their identity source. If your user is connected to an external identity provider or [AD Connector](#), user password resets must be done from the external identity provider or connected Active Directory.

- If your identity source is an **IAM Identity Center directory**, see [Password requirements when managing identities in IAM Identity Center](#).
- If your identity source is an **AWS Managed Microsoft AD**, see [Password requirements when resetting a password in AWS Managed Microsoft AD](#).

To reset your password to the AWS access portal

1. Open a web browser and go to the sign-in page for your AWS access portal.

If you don't have your AWS access portal URL, check your email. You should have been emailed an invitation to join AWS IAM Identity Center that includes a specific sign-in URL to the AWS access portal. Alternatively, your administrator might have directly provided you with a one-time password and the AWS access portal URL. If you can't locate this information, ask your administrator to send it to you.

For more information about signing into the AWS access portal, see [Sign in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

2. Enter your **Username**, and then choose **Next**.
3. Under **Password**, choose **Forgot password**.

Verify your **Username** and enter the characters for the provided image to confirm that you're not a robot. Then choose **Next**. You might need to disable ad blocker software if you can't enter characters.

4. A message appears to confirm that a reset password email was sent. Choose **Continue**.
5. You'll receive an email from `no-reply@signin.aws` with the subject **Password reset requested**. In your email, choose **Reset password**.
6. On the **Reset password** page, verify your **Username**, specify a new password for the AWS access portal, and then choose **Set new password**.
7. You'll receive an email from `no-reply@signin.aws` with the subject line **Password updated**.

Note

An administrator can reset your password by either sending an email to you with instructions for resetting your password or generating a one-time password and sharing it with you. If you're an administrator, see [Reset the IAM Identity Center user password for an end user](#).

Getting IAM Identity Center user credentials for the AWS CLI or AWS SDKs

You can access AWS services programmatically by using the AWS Command Line Interface or AWS Software Development Kits (SDKs) with user credentials from IAM Identity Center. This topic describes how to get temporary credentials for a user in IAM Identity Center.

The AWS access portal provides IAM Identity Center users with single-sign on access to their AWS accounts and cloud applications. After you sign in to the AWS access portal as an IAM Identity Center user, you can get temporary credentials. You can then use the credentials, also referred to as IAM Identity Center user credentials, in the AWS CLI or AWS SDKs to access resources in an AWS account.

If you're using the AWS CLI to access AWS services programmatically, you can use the procedures in this topic to initiate access to the AWS CLI. For information about the AWS CLI, see the [AWS Command Line Interface User Guide](#).

If you're using the AWS SDKs to access AWS services programmatically, following the procedures in this topic also directly establishes authentication for the AWS SDKs. For information about the AWS SDKs, see the [AWS SDKs and Tools Reference Guide](#).

Note

Users in IAM Identity Center are different than [IAM users](#). IAM users are granted long-term credentials to AWS resources. Users in IAM Identity Center are granted temporary credentials. We recommend that you use temporary credentials as a security best practice for accessing your AWS accounts because these credentials are generated every time you sign in.

Prerequisites

To get temporary credentials for your IAM Identity Center user, you'll need the following:

- **An IAM Identity Center user** – You'll sign in to the AWS access portal as this user. You or your administrator might create this user. For information about how to enable IAM Identity Center and create an IAM Identity Center user, see [Get started with common tasks in IAM Identity Center](#).
- **User access to an AWS account** – To grant an IAM Identity Center user permission to retrieve their temporary credentials, you or an administrator must assign the IAM Identity Center user to a [permission set](#). Permission sets are stored in IAM Identity Center and define the level of access that an IAM Identity Center user has to an AWS account. If your administrator created the IAM Identity Center user for you, ask them to add this access for you. For more information, see [Assign user access to AWS accounts](#).
- **AWS CLI installed** – To use the temporary credentials, you must install the AWS CLI. For instructions, see [Installing or updating the latest version of the AWS CLI](#) in the *AWS CLI User Guide*.

Considerations

Before you complete the steps to get temporary credentials for your IAM Identity Center user, keep the following considerations in mind:

- **IAM Identity Center creates IAM roles** – When you assign a user in IAM Identity Center to a permission set, IAM Identity Center creates a corresponding IAM role from the permission set.

IAM roles created by permission sets differ from IAM roles created in AWS Identity and Access Management in the following ways:

- IAM Identity Center owns and secures the roles that are created by permission sets. Only IAM Identity Center can modify these roles.
- Only users in IAM Identity Center can assume the roles that correspond to their assigned permission sets. You can't assign permission set access to IAM users, IAM federated users, or service accounts.
- You can't modify a role trust policy on these roles to allow access to [principals](#) outside of IAM Identity Center.

For information about how to get temporary credentials for a role that you create in IAM, see [Using temporary security credentials with the AWS CLI](#) in the *AWS Identity and Access Management User Guide*.

- **You can set the session duration for permission sets** – After you sign in to the AWS access portal, the permission set to which your IAM Identity Center user is assigned appears as an available role. IAM Identity Center creates a separate session for this role. This session can be from one to 12 hours, depending the session duration configured for the permission set. The default session duration is one hour. For more information, see [Set session duration for AWS accounts](#).

Getting and refreshing temporary credentials

You can get and refresh temporary credentials for your IAM Identity Center user automatically or manually.

Topics

- [Automatic credential refresh \(recommended\)](#)
- [Manual credential refresh](#)

Automatic credential refresh (recommended)

Automatic credential refresh uses the Open ID Connect (OIDC) Device Code Authorization standard. With this method, you initiate access directly by using the `aws configure sso` command in the AWS CLI. You can use this command to automatically access any role that's associated with any permission set that you're assigned to for any AWS account.

To access the role created for your IAM Identity Center user, run the `aws configure sso` command, and then authorize the AWS CLI from a browser window. As long as you have an active AWS access portal session, the AWS CLI automatically retrieves temporary credentials and refreshes the credentials automatically.

For more information, see [Configure your profile with the `aws configure sso` wizard](#) in the *AWS Command Line Interface User Guide*.

To get temporary credentials that automatically refresh

1. Sign in to the AWS access portal by using the specific sign-in URL provided by your administrator. If you created the IAM Identity Center user, AWS sent an email invitation that includes your sign-in URL. For more information, see [Sign in to the AWS access portal](#) in the *AWS Sign-In User Guide*.
2. In the **Accounts** tab, locate the AWS account from which you want to retrieve credentials. When you choose the account, the account name, account ID, and email address associated with the account appear.

Note

If you don't see any **AWS accounts** listed, it's likely that you've not yet been assigned to a permission set for that account. In this case, contact your administrator and ask them to add this access for you. For more information, see [Assign user access to AWS accounts](#).

3. Below the name of the account, the permission set to which your IAM Identity Center user is assigned appears as an available role. For example, if your IAM Identity Center user is assigned to the **PowerUserAccess** permission set for the account, the role appears in the AWS access portal as **PowerUserAccess**.
4. Depending on your option next to the role name, either choose **Access keys** or choose **Command line or programmatic access**.
5. In the **Get credentials** dialog box, choose either **macOS and Linux**, **Windows**, or **PowerShell**, depending on the operating system on which you installed the AWS CLI.
6. Under **AWS IAM Identity Center credentials (Recommended)**, your SSO Start URL and SSO Region are displayed. These values are required to configure both an IAM Identity Center enabled profile and `sso-session` to your AWS CLI. To complete this configuration, follow

the instructions in [Configure your profile with the aws configure sso wizard](#) in the *AWS Command Line Interface User Guide*.

Continue using the AWS CLI as necessary for your AWS account until the credentials have expired.

Manual credential refresh

You can use the manual credential refresh method to get temporary credentials for a role that's associated with a specific permission set in a specific AWS account. To do so, you copy and paste the required commands for the temporary credentials. With this method, you must refresh the temporary credentials manually.

You can run AWS CLI commands until your temporary credentials expire.

To get credentials that you manually refresh

1. Sign in to the AWS access portal by using the specific sign-in URL provided by your administrator. If you created the IAM Identity Center user, AWS sent an email invitation that includes your sign-in URL. For more information, see [Sign in to the AWS access portal](#) in the *AWS Sign-In User Guide*.
2. In the **Accounts** tab, locate the AWS account from which you want to retrieve access credentials and expand it to show the IAM role name (for example **Administrator**). Depending on your option next to the IAM role name, either choose **Access keys** or choose **Command line or programmatic access**.

Note

If you don't see any **AWS accounts** listed, it's likely that you've not yet been assigned to a permission set for that account. In this case, contact your administrator and ask them to add this access for you. For more information, see [Assign user access to AWS accounts](#).

3. In the **Get credentials** dialog box, choose **MacOS and Linux, Windows, or PowerShell**, depending on the operating system on which you installed the AWS CLI.
4. Choose any of the following options:
 - **Option 1: Set AWS environment variables**

Choose this option to override all credential settings, including any settings in the `credentials` files and `config` files. For more information, see [Environment variables to configure the AWS CLI](#) in the *AWS CLI User Guide*.

To use this option, copy the commands to your clipboard, paste the commands into your AWS CLI terminal window, and then press **Enter** to set the required environment variables.

- **Option 2: Add a profile to your AWS credentials file**

Choose this option to run commands with different sets of credentials.

To use this option, copy the commands to your clipboard, and then paste the commands into your shared AWS `credentials` file to set up a new named profile. For more information, see [Shared config and credentials files](#) in the *AWS SDKs and Tools Reference Guide*. To use this credential, specify the `--profile` option in your AWS CLI command. This affects all environments that use the same credential file.

- **Option 3: Use individual values in your AWS service client**

Choose this option to access AWS resources from an AWS service client. For more information, see [Tools to Build on AWS](#).

To use this option, copy the values to your clipboard, paste the values into your code, and assign them to the appropriate variables for your SDK. For more information, see the documentation for your specific SDK API.

Creating shortcut links to AWS Management Console destinations

Shortcut links created in the AWS access portal take IAM Identity Center users to a specific destination in the AWS Management Console, with a specific permission set, and in a specific AWS account.

Shortcut links save time for you and your collaborators. Instead of navigating to a desired destination URL in the AWS Management Console (for example, an Amazon S3 bucket instance page) through multiple pages, including AWS access portal, you can use a shortcut link to get to the same destination automatically.

Shortcut link destination options

Shortcut links have three destination options, listed here by priority:

- (Optional) Any destination URL in the AWS Management Console specified in the shortcut link. For example, the Amazon S3 bucket instance page.
- (Optional) Administrator-configured relay state URL for the permission set in question. For more information about setting the relay state, see [Set relay state for quick access to the AWS Management Console](#).
- AWS Management Console home. The default destination if you don't specify one.

Note

Automatic navigation to a destination is successful only when you're authenticated with IAM Identity Center and have the necessary permission set assigned for the AWS account and destination URL.

The AWS access portal includes a **Create shortcut** button that helps you create a shareable shortcut link. If you plan to specify a destination URL (the first option in the previous list), you can copy the URL to a clipboard to share it.

Create a shortcut link in the AWS access portal

1. While signed into the AWS access portal, choose the **Accounts** tab and then choose the **Create shortcut** button.
2. In the dialog box:
 - a. Choose an AWS account using the account ID or account name. As you type, a drop-down menu displays matching account IDs and names that you can access. You can choose only an account to which you have access.
 - b. Optionally choose an IAM role from the drop-down list. These are the permission sets assigned to you for the selected account. If you omit choosing the role, users are prompted to select one assigned to them for the chosen account when using the shortcut link.

Note

You can't grant new access with shortcut links. Shortcut links work only with the permission sets already assigned to the user. If the user doesn't have the necessary

permission sets assigned for the account and destination URL, they're denied access.

- c. Optionally enter the AWS access portal destination URL. If you omit entering a URL, the destination is automatically determined when using the shortcut link, based on the previously-mentioned shortcut link destination options.
- d. Your shortcut link generates at the bottom of the dialog box, based on your input. Choose the **Copy URL** button. You can now create a bookmark with the copied shortcut link or share it with your collaborators who have access to the same account with the same permission set or another sufficient permission set.

Constructing secure AWS Management Console shortcut links with URL encoding

All parameter values of the URL, including the account ID, permission set name, and destination URL, must be URL-encoded.

Shortcut links extend the AWS access portal URL with the following path:

```
/#/console?
```

```
account_id=[account_ID]&role_name=[permission_set_name]&destination=[destination]
```

The full URL in the classic AWS partition follows this pattern:

```
https://[your_subdomain].awsapps.com/start/#/console?
```

```
account_id=[account_ID]&role_name=[permission_set_name]&destination=[destination]
```

Here's an example shortcut link that signs a user into account 123456789012 with the S3FullAccess permission set, and takes them to the S3 console home page:

- `https://example.awsapps.com/start/#/console?account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F%2Fconsole.aws.amazon.com%2Fs3%2Fhome`
- (AWS GovCloud (US) Region) `https://start.us-gov-west-1.us-gov-home.awsapps.com/directory/example/#/console?account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F%2Fconsole.amazonaws-us-gov.com%2Fs3%2Fhome`

Registering your device for MFA

Use the following procedure within the AWS access portal to register your new device for multi-factor authentication (MFA).

Note

We recommend that you first download the appropriate Authenticator app onto your device before starting the steps in this procedure. For a list of apps that you can use for MFA devices, see [Virtual authenticator apps](#).

To register your device for use with MFA

1. Sign in to your AWS access portal. For more information, see [Signing in to the AWS access portal](#).
2. Near the top-right of the page, choose **MFA devices**.
3. On the **Multi-factor authentication (MFA) devices** page, choose **Register device**.

Note

If the **Register MFA device** option is grayed out, contact your administrator for assistance with registering your device.

4. On the **Register MFA device** page, select one of the following MFA device types, and follow the instructions:
 - **Authenticator app**
 1. On the **Set up the authenticator app** page, you might notice configuration information for the new MFA device, including a QR code graphic. The graphic is a representation of the secret key that's available for manual entry on devices that don't support QR codes.
 2. Using the physical MFA device, do the following:
 - a. Open a compatible MFA authenticator app. For a list of tested apps that you can use with MFA devices, see [Virtual authenticator apps](#). If the MFA app supports multiple accounts (multiple MFA devices), choose the option to create a new account (a new MFA device).

- b. Determine whether the MFA app supports QR codes, and then do one of the following on the **Set up the authenticator app** page:
 - i. Choose **Show QR code**, and then use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to **Scan code**. Then use the device's camera to scan the code.
 - ii. Choose **show secret key**, and then enter that secret key into your MFA app.

 **Important**

When you configure an MFA device for IAM Identity Center, we recommend that you save a copy of the QR code or secret key *in a secure place*. This can help if you lose the phone or have to reinstall the MFA authenticator app. If either of those things happen, you can quickly reconfigure the app to use the same MFA configuration.

3. On the **Set up the authenticator app** page, under **Authenticator code**, enter the one-time password that currently appears on the physical MFA device.


 **Important**

Submit your request immediately after generating the code. If you generate the code and then wait too long to submit the request, the MFA device is successfully associated with your user, but the MFA device is out of sync. This happens because time-based one-time passwords (TOTP) expire after a short period of time. If this happens, you can sync the device again.

4. Choose **Assign MFA**. The MFA device can now start generating one-time passwords and is now ready for use with AWS.

- **Security key or Built-in authenticator**

1. On the **Register your user's security key** page, follow the instructions provided by your browser or platform.

 **Note**

The experience varies based on the browser or platform. After your device is successfully registered, you can associate a friendly display name with your newly

enrolled device. To to change the name, choose **Rename**, enter the new name, and then choose **Save**.

Customizing the AWS access portal URL

By default, you can access the AWS access portal by using a URL that follows this format: `d-xxxxxxxxxx.awsapps.com/start`. You can customize the URL as follows: `your_subdomain.awsapps.com/start`.

Important

If you change the AWS access portal URL, you can't edit it later.

To customize your URL

1. Open the AWS IAM Identity Center console at <https://console.aws.amazon.com/singlesignon/>.
2. In the IAM Identity Center console, choose **Dashboard** in the navigation pane and locate the **Settings summary** section.
3. Choose the **Customize** button below your AWS access portal URL.

Note

If the **Customize** button doesn't display, it means that the AWS access portal has already been customized. Customizing the AWS access portal URL is a one-time operation that can't be reversed.

4. Enter your desired subdomain name and choose **Save**.

You can now sign in to the AWS Console through your AWS access portal with your customized URL.

Multi-factor authentication for Identity Center users

Multi-factor authentication (MFA) provides a simple and secure way to add an extra layer of protection on top of the default authentication mechanism of user name and password.

When administrators enable MFA, users must sign in to the AWS access portal with two factors:

- Their user name and password. This is the first factor and is something users know.
- Either a code, security key, or biometrics. This is the second factor and is something users have (possession) or are (biometric). The second factor might be either an authentication code generated from their mobile device, a security key connected to their computer, or user's biometric scan.

Together, these multiple factors provide increased security by preventing unauthorized access to your AWS resources unless a valid MFA challenge has been successfully completed.

Each user can register up to two virtual authenticator apps, which are one-time password authenticator applications installed on your mobile device or tablet, and six FIDO authenticators, which include built-in authenticators and security keys, for a total of **eight** MFA devices. Learn more about [Available MFA types for IAM Identity Center](#).

Important

As a security best practice, we strongly recommend that you enable MFA.

Topics

- [Available MFA types for IAM Identity Center](#)
- [Configure MFA devices in IAM Identity Center](#)
- [Manage MFA devices in IAM Identity Center](#)

Available MFA types for IAM Identity Center

Multi-factor authentication (MFA) is a simple and effective mechanism to enhance the security of your users. A user's first factor — their password — is a secret that they memorize, also known as a knowledge factor. Other factors can be possession factors (something you have, such as a security key) or inherence factors (something you are, such as a biometric scan). We strongly recommend that you configure MFA to add an additional layer of security to your account.

IAM Identity Center MFA supports the following device types. All MFA types are supported for both browser-based console access as well as using the AWS CLI v2 with IAM Identity Center.

- [FIDO2 authenticators](#), including built-in authenticators and security keys
- [Virtual authenticator apps](#)
- Your own [RADIUS MFA](#) implementation connected through AWS Managed Microsoft AD

A user can have up to **eight** MFA devices, which include up to two virtual authenticator apps and six FIDO authenticators, registered to one account. You can also configure MFA enablement settings to require MFA each time your users sign in or to enable trusted devices that don't require MFA at every sign-in. For more information about how to configure MFA types for your users, see [Choose MFA types](#) and [Configure MFA device enforcement](#).

FIDO2 authenticators

[FIDO2](#) is a standard that includes CTAP2 and [WebAuthn](#) and is based on public key cryptography. FIDO credentials are phishing-resistant because they are unique to the website that the credentials were created such as AWS.

AWS supports the two most common form factors for FIDO authenticators: built-in authenticators and security keys. See below for more information about the most common types of FIDO authenticators.

Topics

- [Built-in authenticators](#)
- [Security keys](#)
- [Password managers, passkey providers, and other FIDO authenticators](#)

Built-in authenticators

Many modern computers and mobile phones have built-in authenticators, such as TouchID on Macbook or a Windows Hello-compatible camera. If your device has a FIDO-compatible built-in authenticator, you can use your fingerprint, face, or device pin as a second factor.

Security keys

Security keys are FIDO-compatible external hardware authenticators that you can purchase and connect to your device through USB, BLE, or NFC. When you're prompted for MFA, you simply complete a gesture with the key's sensor. Some examples of security keys include YubiKeys and

Fetian keys, and the most common security keys create device-bound FIDO credentials. For a list of all FIDO-certified security keys, see [FIDO Certified Products](#).

Password managers, passkey providers, and other FIDO authenticators

Multiple third party providers support FIDO authentication in mobile applications, as features in password managers, smart cards with a FIDO mode, and other form factors. These FIDO-compatible devices can work with IAM Identity Center, but we recommend that you test a FIDO authenticator yourself before enabling this option for MFA.

Note

Some FIDO authenticators can create discoverable FIDO credentials known as passkeys. Passkeys may be bound to the device that creates them, or they may be syncable and backed up to a cloud. For example, you can register a passkey using Apple Touch ID on a supported Macbook, and then log in to a site from a Windows laptop using Google Chrome with your passkey in iCloud by following the on-screen prompts at sign-in. For more information about which devices support syncable passkeys and current passkey interoperability between operating systems and browsers, see [Device Support at passkeys.dev](#), a resource maintained by the FIDO Alliance And World Wide Web Consortium (W3C).

Virtual authenticator apps

Authenticator apps are essentially one-time password (OTP)–based third party-authenticators. You can use an authenticator application installed on your mobile device or tablet as an authorized MFA device. The third-party authenticator application must be compliant with RFC 6238, which is a standards-based time-based one-time password (TOTP) algorithm capable of generating six-digit authentication codes.

When prompted for MFA, users must enter a valid code from their authenticator app within the input box presented. Each MFA device assigned to a user must be unique. Two authenticator apps can be registered for any given user.

Tested authenticator apps

Any TOTP-compliant application will work with IAM Identity Center MFA. The following table lists well-known third-party authenticator apps to choose from.

Operating system	Tested authenticator app
Android	Authy , Duo Mobile , Microsoft Authenticator , Google Authenticator
iOS	Authy , Duo Mobile , Microsoft Authenticator , Google Authenticator

RADIUS MFA

[Remote Authentication Dial-In User Service \(RADIUS\)](#) is an industry-standard client-server protocol that provides authentication, authorization, and accounting management so users can connect to network services. AWS Directory Service includes a RADIUS client that connects to the RADIUS server upon which you have implemented your MFA solution. For more information, see [Enable Multi-Factor Authentication for AWS Managed Microsoft AD](#).

You can use either RADIUS MFA or MFA in IAM Identity Center for user sign-ins to the user portal, but not both. MFA in IAM Identity Center is an alternative to RADIUS MFA in cases where you want AWS native two-factor authentication for access to the portal.

When you enable MFA in IAM Identity Center, your users need an MFA device to sign in to the AWS access portal. If you had previously used RADIUS MFA, enabling MFA in IAM Identity Center effectively overrides RADIUS MFA for users who sign in to the AWS access portal. However, RADIUS MFA continues to challenge users when they sign in to all other applications that work with AWS Directory Service, such as Amazon WorkDocs.

If your MFA is **Disabled** on the IAM Identity Center console and you have configured RADIUS MFA with AWS Directory Service, RADIUS MFA governs AWS access portal sign-in. This means that IAM Identity Center falls back to RADIUS MFA configuration if MFA is disabled.

Configure MFA devices in IAM Identity Center

Learn about configuring MFA devices in IAM Identity Center.

Considerations before enabling MFA in IAM Identity Center

Before you enable MFA, consider the following:

- Users are encouraged to register multiple backup authenticators for all enabled MFA types. This practice can prevent loss of access in case of a broken or misplaced MFA device.
- Don't choose the **Require Them to Provide a One-Time Password Sent by Email** option if your users must sign in to the AWS access portal to access their email. For example, your users might use Microsoft 365 in the AWS access portal to read their email. In this case, users won't be able to retrieve the verification code and would be unable to sign in to the AWS access portal. For more information, see [Configure MFA device enforcement](#).
- If you're already using RADIUS MFA that you configured with AWS Directory Service, you don't need to enable MFA within IAM Identity Center. MFA in IAM Identity Center is an alternative to RADIUS MFA for Microsoft Active Directory users of IAM Identity Center. For more information, see [RADIUS MFA](#).
- You can use MFA capabilities in IAM Identity Center when your identity source is configured with IAM Identity Center's identity store, AWS Managed Microsoft AD, or AD Connector. MFA in IAM Identity Center is currently not supported for [external identity providers](#).

Topics

- [Enable MFA in IAM Identity Center](#)
- [Choose MFA types](#)
- [Configure MFA device enforcement](#)
- [Allow users to register their own MFA devices](#)

Enable MFA in IAM Identity Center

You can enable secure access to the AWS access portal, IAM Identity Center integrated apps, and the AWS CLI by enabling multi-factor authentication (MFA).

Topics

- [Prompt users for MFA](#)
- [Disable MFA for your IAM Identity Center directory](#)

Prompt users for MFA

Use the following steps to enable MFA in the IAM Identity Center console. Before you begin, we recommend that you understand the [Available MFA types for IAM Identity Center](#).

Note

If you're using an external IdP, the **Multi-factor authentication** section won't be available. Your external IdP manages MFA settings, rather than IAM Identity Center managing them.

To enable MFA

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. In the **Multi-factor authentication** section, choose **Configure**.
5. On the **Configure multi-factor authentication** page, under **Prompt users for MFA**, choose one of the following authentication modes based on the level of security that your business needs:
 - **Only when their sign-in context changes (context-aware)**

In this mode (the default), IAM Identity Center provides users the option to trust their device during sign-in. After a user indicates that they want to trust a device, IAM Identity Center prompts the user for MFA once and analyzes the sign-in context (such as device, browser, and location) for the user's subsequent sign-ins. For subsequent sign-ins, IAM Identity Center determines if the user is signing in with a previously trusted context. If the user's sign-in context changes, IAM Identity Center prompts the user for MFA in addition to their email address and password credentials.


This mode provides ease of use for users who frequently sign in from their workplace, so they don't need to complete MFA on every sign-in. They are only prompted for MFA if their sign-in context changes.

- **Every time they sign in (always-on)**

In this mode, IAM Identity Center requires that users with a registered MFA device will be prompted every time they sign in. You should use this mode if you have organizational or compliance policies that require your users to complete MFA every time they sign in to the AWS access portal. For example, PCI DSS strongly recommends MFA during every sign-in to access applications that support high-risk payment transactions.

- **Never (disabled)**

While in this mode, all users will sign in with their standard user name and password only. Choosing this option disables IAM Identity Center MFA.

 **Note**

If you are already using RADIUS MFA with AWS Directory Service, and want to continue using it as your default MFA type, then you can leave the authentication mode as disabled to bypass MFA capabilities in IAM Identity Center. Changing from **Disabled** mode to **Context-aware** or **Always-on** mode will override the existing RADIUS MFA settings. For more information, see [RADIUS MFA](#).

6. Choose **Save changes**.

Related Topics

- [Choose MFA types](#)
- [Configure MFA device enforcement](#)
- [Allow users to register their own MFA devices](#)

Disable MFA for your IAM Identity Center directory

When you disable multi-factor authentication (MFA) for your IAM Identity Center directory, it allows users to sign in with their standard user name and password only. While MFA is disabled for your Identity Center directory for users, you can't manage MFA devices in their user details, and Identity Center directory users can't manage MFA devices from the AWS access portal.

To disable MFA for your IAM Identity Center directory

 **Important**

The instructions in this section apply to [AWS IAM Identity Center](#). They do not apply to [AWS Identity and Access Management](#) (IAM). IAM Identity Center users, groups, and user credentials are different from IAM users, groups, and IAM user credentials. If you are looking for instructions on deactivating MFA for IAM users, see [Deactivating MFA devices](#) in the *AWS Identity and Access Management User Guide*.

1. Open the [IAM Identity Center console](#).

2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. In the **Multi-factor authentication** section, choose **Configure**.
5. On the **Configure multi-factor authentication** page, in the **Prompt users for MFA** section, choose the **Never (disabled)** radio button.
6. Choose **Save changes**.

Choose MFA types

Use the following procedure to choose the device types your users can authenticate with when prompted for MFA in the AWS access portal.

To configure MFA types for your users

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. In the **Multi-factor authentication** section, choose **Configure**.
5. On the **Configure multi-factor authentication** page, under **Users can authenticate with these MFA types** choose one of the following MFA types based on your business needs. For more information, see [Available MFA types for IAM Identity Center](#).
 - **FIDO2 authenticators, including built-in authenticators and security keys**
 - **Virtual authenticator apps**
6. Choose **Save changes**.

Configure MFA device enforcement

Use the following procedure to determine whether your users must have a registered MFA device when signing in to the AWS access portal.

To configure MFA device enforcement for your users

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.


4. In the **Multi-factor authentication** section, choose **Configure**.
5. On the **Configure multi-factor authentication** page, under **If a user does not yet have a registered MFA device** choose one of the following choices based on your business needs:

- **Require them to register an MFA device at sign in**

This is the default setting when you first configure MFA for IAM Identity Center. Use this option when you want to require users who do not yet have a registered MFA device, to self-enroll a device during sign-in following a successful password authentication. This allows you to secure your organization's AWS environments with MFA without having to individually enroll and distribute authentication devices to your users. During self-enrollment, your users can register any device from the available [Available MFA types for IAM Identity Center](#) you've previously enabled. After completing registration, users have the option to give their newly enrolled MFA device a friendly name, after which IAM Identity Center redirects the user to their original destination. If the user's device is lost or stolen, you can simply remove that device from their account, and IAM Identity Center will require them to self-enroll a new device during their next sign-in.

- **Require them to provide a one-time password sent by email to sign in**

Use this option when you want to have verification codes sent to users by email. Because email is not bound to a specific device, this option does not meet the bar for industry-standard multi-factor authentication. But it does improve security over having a password alone. Email verification will only be requested if a user has not registered an MFA device. If the **Context-aware** authentication method has been enabled, the user will have the opportunity to mark the device on which they receive the email as trusted. Afterward they will not be required to verify an email code on future logins from that device, browser, and IP address combination.

 **Note**

If you are using Active Directory as your IAM Identity Center enabled identity source, the email address will always be based on the Active Directory email attribute. Custom Active Directory attribute mappings will not override this behavior.

- **Block their sign-in**

Use the **Block Their Sign-In** option when you want to enforce MFA use by every user before they can sign in to AWS.

⚠ Important

If your authentication method is set to **Context-aware** a user might select the **This is a trusted device** check box on the sign-in page. In that case, that user will not be prompted for MFA even if you have the **Block their sign in** setting enabled. If you want these users to be prompted, change your authentication method to **Always On**.

- **Allow them to sign in**

Use this option to indicate that MFA devices are not required in order for your users to sign in to the AWS access portal. Users who chose to register MFA devices will still be prompted for MFA.

6. Choose **Save changes**.

Allow users to register their own MFA devices

Use the following procedure to allow your users to self-register their own MFA devices.

To allow users to register their own MFA devices

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. In the **Multi-factor authentication** section, choose **Configure**.
5. On the **Configure multi-factor authentication** page, under **Who can manage MFA devices**, choose **Users can add and manage their own MFA devices**.
6. Choose **Save changes**.

📘 Note

After you set up self-registration for your users, you might want to send them a link to the procedure [Registering your device for MFA](#). This topic provides instructions on how to set up their own MFA devices.

Manage MFA devices in IAM Identity Center

The following topics provide instructions for managing MFA devices in IAM Identity Center.

Topics

- [Register an MFA device](#)
- [Renaming and deleting MFA devices in IAM Identity Center](#)

Register an MFA device

Use the following procedure to set up a new MFA device for access by a specific user in the IAM Identity Center console. You must have physical access to the user's MFA device in order to register it. For example, if you configure MFA for a user who will use an MFA device running on a smartphone, you'll need physical access to the smartphone to complete the registration process. Alternatively, you can allow users to configure and manage their own MFA devices. For more information, see [Allow users to register their own MFA devices](#).

To register an MFA device

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Users**. Choose a user in the list. Don't select the checkbox next to the user for this step.
3. On the user details page, choose the **MFA devices** tab, and then choose **Register MFA device**.
4. On the **Register MFA device** page, select one of the following MFA device types, and follow the instructions:
 - **Authenticator app**
 1. On the **Set up the authenticator app** page, IAM Identity Center displays configuration information for the new MFA device, including a QR code graphic. The graphic is a representation of the secret key that is available for manual entry on devices that do not support QR codes.
 2. Using the physical MFA device, do the following:
 - a. Open a compatible MFA authenticator app. For a list of tested apps that you can use with MFA devices, see [Virtual authenticator apps](#). If the MFA app supports multiple accounts (multiple MFA devices), choose the option to create a new account (a new MFA device).

- b. Determine whether the MFA app supports QR codes, and then do one of the following on the **Set up the authenticator app** page:
 - i. Choose **Show QR code**, and then use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to **Scan code**. Then use the device's camera to scan the code.
 - ii. Choose **show secret key**, and then type that secret key into your MFA app.

 **Important**

When you configure an MFA device for IAM Identity Center, we recommend that you save a copy of the QR code or secret key *in a secure place*. This can help if the assigned user loses the phone or has to reinstall the MFA authenticator app. If either of those things happen, you can quickly reconfigure the app to use the same MFA configuration. This avoids the need to create a new MFA device in IAM Identity Center for the user.

3. On the **Set up the authenticator app** page, under **Authenticator code**, type the one-time password that currently appears on the physical MFA device.

 **Important**

Submit your request immediately after generating the code. If you generate the code and then wait too long to submit the request, the MFA device is successfully associated with the user. But the MFA device is out of sync. This happens because time-based one-time passwords (TOTP) expire after a short period of time. If this happens, you can resync the device.

4. Choose **Assign MFA**. The MFA device can now start generating one-time passwords and is now ready for use with AWS.

- **Security key**

1. On the **Register your user's security key** page, follow the instructions given to you by your browser or platform.

Note

The experience here varies based on the different operating systems and browsers, so please follow the instructions displayed by your browser or platform. After your user's device has been successfully registered, you will be given the option to associate a friendly display name to your user's newly enrolled device. If you want to change this, choose **Rename**, enter the new name, and then choose **Save**. If you have enabled the option to allow users to manage their own devices, the user will see this friendly name in the AWS access portal.

Renaming and deleting MFA devices in IAM Identity Center

Use the following procedures to rename or delete a user's MFA device.

To rename an MFA device

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Users**. Choose the user in the list. Don't select the checkbox next to the user for this step.
3. On the user details page, choose the **MFA devices** tab, select the device, and then choose **Rename**.
4. When prompted, enter the new name and then choose **Rename**.

To delete an MFA device

1. Open the [IAM Identity Center console](#).
2. In the left navigation pane, choose **Users**. Choose the user in the list.
3. On the user details page, choose the **MFA devices** tab, select the device, and then choose **Delete**.
4. To confirm, type **DELETE**, and then choose **Delete**.

Manage access to AWS accounts

AWS IAM Identity Center is integrated with AWS Organizations, which enables you to centrally manage permissions across multiple AWS accounts without configuring each of your accounts manually. You can define permissions and assign these permissions to workforce users to control their access to specific AWS accounts using an [organization instance](#) of IAM Identity Center. [Account instances](#) of IAM Identity Center don't support account access.

AWS account types




There are two types of AWS accounts in AWS Organizations:

- **Management account** - The AWS account that is used to create the organization.
- **Member accounts** - The rest of the AWS accounts that belong to an organization.



















For more information about AWS account types, see [AWS Organizations Terminology and Concepts](#) in the *AWS Organizations User Guide*.

You can also choose to register a member account as a *delegated administrator* for IAM Identity Center. Users in this account can perform most IAM Identity Center administrative tasks. For more information, see [Delegated administration](#).

For each task and account type, the following table indicates whether the IAM Identity Center administrative task can be performed by users in the account.

IAM Identity Center administrative tasks	Member account	Delegated administrator account	Management account
Read users or groups (reading the group itself and the group's membership)	 Yes	 Yes	 Yes

IAM Identity Center administrative tasks	Member account	Delegated administrator account	Management account
Add, edit, or delete users or groups	 No	 Yes	 Yes
Enable or disable user access	 No	 Yes	 Yes
Enable, disable, or manage incoming attributes	 No	 Yes	 Yes
Change or manage identity sources	 No	 Yes	 Yes
Create, edit, or delete customer managed applications	 No	 Yes	 Yes
Create, edit, or delete AWS managed applications	 Yes	 Yes	 Yes
Configure MFA	 No	 Yes	 Yes

IAM Identity Center administrative tasks	Member account	Delegated administrator account	Management account
Manage permission sets not provisioned in the management account	 No	 Yes	 Yes
Manage permission sets provisioned in the management account	 No	 No	 Yes
Enable IAM Identity Center	 No	 No	 Yes
Delete IAM Identity Center configuration	 No	 No	 Yes
Enable or disable user access in the management account	 No	 No	 Yes
Register or deregister a member account as a delegated administrator	 No	 No	 Yes

Assigning AWS account access

You can use *permission sets* to simplify how you assign users and groups in your organization access to AWS accounts. Permission sets are stored in IAM Identity Center and define the level of access that users and groups have to an AWS account. You can create a single permission set and assign it to multiple AWS accounts within your organization. You can also assign multiple permission sets to the same user.

For more information about permission sets, see [Create, manage, and delete permission sets](#).

Note

You can also assign your users single sign-on access to applications. For information, see [Manage access to applications](#).

End-user experience

The *AWS access portal* provides IAM Identity Center users with single sign-on access to all their assigned AWS accounts and applications through a web portal. The AWS access portal is different from the [AWS Management Console](#), which is a collection of service consoles for managing AWS resources.

When you create a permission set, the name that you specify for the permission set appears in the AWS access portal as an available role. Users sign in to the AWS access portal, choose an AWS account, and then choose the role. After they choose the role, they can access AWS services by using the AWS Management Console or retrieve temporary credentials to access AWS services programmatically.

To open the AWS Management Console or retrieve temporary credentials to access AWS programmatically, users complete the following steps:

1. Users open a browser window and use the sign-in URL that you provide to navigate to the AWS access portal.
2. Using their directory credentials, they sign in to the AWS access portal.
3. After authentication, on the AWS access portal page, they choose the **Accounts** tab to display the list of AWS accounts to which they have access.
4. Users then choose the AWS account that they want to use.

5. Below the name of the AWS account, any permission sets to which users are assigned appear as available roles. For example, if you assigned user `john_stiles` to the `PowerUser` permission set, the role displays in the AWS access portal as `PowerUser/john_stiles`. Users who are assigned multiple permission sets choose which role to use. Users can choose their role to access the AWS Management Console.
6. In addition to the role, AWS access portal users can retrieve temporary credentials for command line or programmatic access by choosing **Access keys**.

For step-by-step guidance that you can provide to your workforce users, see [Using the AWS access portal](#) and [Getting IAM Identity Center user credentials for the AWS CLI or AWS SDKs](#).

Enforcing and limiting access

When you enable IAM Identity Center, IAM Identity Center creates a service-linked role. You can also use service control policies (SCPs).

Delegating and enforcing access

A *service-linked role* is a type of IAM role that is linked directly to an AWS service. After you enable IAM Identity Center, IAM Identity Center can create a service-linked role in each AWS account in your organization. This role provides predefined permissions that allow IAM Identity Center to delegate and enforce which users have single sign-on access to specific AWS accounts in your organization in AWS Organizations. You need to assign one or more users with access to an account, to use this role. For more information, see [Service-linked roles](#) and [Using service-linked roles for IAM Identity Center](#).

Limiting access to the identity store from member accounts

For the identity store service used by IAM Identity Center, users who have access to a member account can use API actions that require **Read** permissions. Member accounts have access to **Read** actions on both the **sso-directory** and **identitystore** namespaces. For more information, see [Actions, resources, and condition keys for AWS IAM Identity Center directory](#) and [Actions, resources, and condition keys for AWS Identity Store](#) in the *Service Authorization Reference*.

To prevent users in member accounts from using API operations in the identity store, you can [attach a service control policy \(SCP\)](#). An SCP is a type of organization policy that you can use to manage permissions in your organization. The following example SCP prevents users in member accounts from accessing any API operation in the identity store.

```
{
  "Sid": "ExplicitlyBlockIdentityStoreAccess",
  "Effect": "Deny",
  "Action": "identitystore:*", "sso-directory:*"],
  "Resource": "*"
}
```

Note

Limiting member accounts' access might impair functionality in IAM Identity Center enabled applications.

For more information, see [Service control policies \(SCPs\)](#) in the *AWS Organizations User Guide*.

Delegated administration

Delegated administration provides a convenient way for assigned users in a registered member account to perform most IAM Identity Center administrative tasks. When you enable IAM Identity Center, your IAM Identity Center instance is created in the management account in AWS Organizations by default. This was originally designed this way so that IAM Identity Center can provision, de-provision, and update roles across all your organization's member accounts. Even though your IAM Identity Center instance must always reside in the management account, you can choose to delegate administration of IAM Identity Center to a member account in AWS Organizations, thereby extending the ability to manage IAM Identity Center from outside the management account.

Enabling delegated administration provides the following benefits:

- Minimizes the number of people who require access to the management account to help mitigate security concerns
- Allows select administrators to assign users and groups to applications and to your organization's member accounts

For more information about how IAM Identity Center works with AWS Organizations, see [Manage access to AWS accounts](#). For additional information and to review an example company scenario showing how to configure delegated administration, see [Getting started with IAM Identity Center delegated administration](#) in the *AWS Security Blog*.

Topics

- [Best practices](#)
- [Prerequisites](#)
- [Register a member account](#)
- [Deregister a member account](#)
- [View which member account has been registered as the delegated administrator](#)

Best practices

Here are some best practices to consider before you configure delegated administration.

- **Grant least privilege to the management account** – Knowing that the management account is a highly privileged account and to adhere to the principal of least privilege, we highly recommend that you restrict access to the management account to as few people as possible. The delegated administrator feature is intended to minimize the number of people who require access to the management account.
- **Create permission sets for use only in the management account** – This makes it easier to administer permission sets tailored just for users accessing your management account and helps to differentiate them from permission sets managed by your delegated administrator account.
- **Consider your Active Directory location** – If you plan on using Active Directory as your IAM Identity Center identity source, locate the directory in the member account where you have enabled the IAM Identity Center delegated administrator feature. If you decide to change the IAM Identity Center identity source from any other source to Active Directory, or change it from Active Directory to any other source, the directory must reside in (be owned by) the IAM Identity Center delegated administrator member account if one exists; otherwise, it must be in the management account.
- **Create user assignments only in the management account** – The delegated administrator can't alter permission sets provisioned in the management account. However, delegated administrators can add, edit, and delete groups and group assignments.

Prerequisites

Before you can register an account as a delegated administrator you must first have the following environment deployed:

- AWS Organizations must be enabled and configured with at least one member account in addition to your default management account.
- If your identity source is set to Active Directory, the [IAM Identity Center configurable AD sync](#) feature must be enabled.

Register a member account

To configure delegated administration, you must first register a member account in your organization as a delegated administrator. Users in that member account who have sufficient permissions will have administrative access to IAM Identity Center. After a member account is successfully registered for delegated administration, it's referred to as the *delegated administrator account*. To learn more about tasks that the delegated administrator account can perform, see [AWS account types](#).

IAM Identity Center supports registering only one member account as a delegated administrator at a time. You can only register a member account while signed in with credentials from the management account.

Use the following procedure to grant administrative access to IAM Identity Center by registering a specific member account in your AWS organization as a delegated administrator.

Important

This operation delegates IAM Identity Center administrative access to admin users in this member account. All users who have sufficient permissions to this delegated administrator account can perform all IAM Identity Center administrative tasks from the account, except for:

- Enabling IAM Identity Center
- Deleting IAM Identity Center configurations
- Managing permission sets provisioned in the management account
- Registering or deregistering other member accounts as delegated administrators
- Enabling or disabling user access in the management account

The delegated administrator can edit group membership.

To register a member account

1. Sign in to the AWS Management Console using the credentials of your management account in AWS Organizations. Management account credentials are required to run the [RegisterDelegatedAdministrator](#) API.
2. Select the Region where IAM Identity Center is enabled, and then open the [IAM Identity Center console](#).
3. Choose **Settings**, and then select the **Management** tab.
4. In the **Delegated administrator** section, choose **Register account**.
5. On the **Register delegated administrator** page, select the AWS account you want to register, and then choose **Register account**.

Deregister a member account

You can only deregister a member account while signed in with credentials from the management account.

Use the following procedure to remove administrative access from IAM Identity Center by deregistering a member account in your AWS organization that had previously been designated as a delegated administrator.

Important

When you deregister an account, you effectively remove the ability for all admin users to manage IAM Identity Center from that account. As a result, they can no longer administer IAM Identity Center identities, access management, authentication, or application access from this account. This operation will not affect any permissions or assignments configured in IAM Identity Center and therefore will have no impact on your end users as they will continue to have access to their apps and AWS accounts from within the AWS access portal.

To deregister a member account

1. Sign in to the AWS Management Console using the credentials of your management account in AWS Organizations. Management account credentials are required to run the [DeregisterDelegatedAdministrator](#) API.

2. Select the Region where IAM Identity Center is enabled, and then open the [IAM Identity Center console](#).
3. Choose **Settings**, and then select the **Management** tab.
4. In the **Delegated administrator** section, choose **Deregister account**.
5. In the **Deregister account** dialog box, review the security implications, and then enter the name of the member account to confirm that you understand.
6. Choose **Deregister account**.

View which member account has been registered as the delegated administrator

Use the following procedure to find which member account in your AWS Organizations has been configured as the delegated administrator for IAM Identity Center.

To view your registered member account

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. In the **Details** section, locate the registered account name under **Delegated administrator**. You can also locate this information by selecting the **Management** tab, and viewing it under the **Delegated administrator** section.

Temporary elevated access for AWS accounts

All access to your AWS account involves some level of privilege. Sensitive operations, such as changing the configuration for a high-value resource, for example, a production environment, require special treatment due to scope and potential impact. Temporary elevated access (also known as just-in-time access) is a way to request, approve, and track the use of a permission to perform a specific task during a specified time. Temporary elevated access supplements other forms of access control, such as permission sets and multi-factor authentication.

AWS IAM Identity Center provides the following options for temporary elevated access management in different business and technical environments:

- **Vendor-managed and supported solutions** – AWS has validated the IAM Identity Center integrations of [select partner offerings](#) and assessed their capabilities against a [common set of](#)

[customer requirements](#). Choose the solution that best aligns with your scenario and follow the provider's guidance to enable the capability with IAM Identity Center.

- **Self-managed and self-supported** – This option provides a starting point if you are interested in temporary elevated access to AWS only and you can deploy, tailor, and maintain the capability by yourself. For more information, see [Temporary elevated access management \(TEAM\)](#).

Validated AWS Security Partners for temporary elevated access

AWS Security Partners use different approaches to address a [common set of temporary elevated access requirements](#). We recommend that you review each partner solution carefully, so that you can choose one that best fits your needs and preferences, including your business, the architecture of your cloud environment, and your budget.

Note

For disaster recovery, we recommend that you [set up emergency access to the AWS Management Console](#) before a disruption occurs.

AWS Identity has validated the capabilities and integration with IAM Identity Center for the following just-in-time offerings by AWS Security Partners:

- **[CyberArk Secure Cloud Access](#)** – Part of the CyberArk Identity Security Platform, this offering provisions on-demand elevated access to AWS and multi-cloud environments. Approvals are addressed through integration with either ITSM or ChatOps tooling. All sessions can be recorded for audit and compliance.
- **[Tenable \(previously Ermetic\)](#)** – The Tenable platform includes provisioning of just-in-time privileged access for administrative operations in AWS and multi-cloud environments. Session logs from all cloud environments, including AWS CloudTrail access logs, are available in a single interface for analysis and audit. The capability integrates with enterprise and developer tools such as Slack and Microsoft Teams.
- **[Okta Access Requests](#)** – Part of Okta Identity Governance, enables you to [configure a just-in-time access request workflow using Okta](#) as an IAM Identity Center external identity provider (IdP) and your IAM Identity Center permission sets.

This list will be updated as AWS validates the capabilities of additional partner solutions and integration of these solutions with IAM Identity Center.

Note

If you are using resource-based policies, Amazon Elastic Kubernetes Service (Amazon EKS), or AWS Key Management Service (AWS KMS), see [Referencing permission sets in resource policies, Amazon EKS Cluster config maps, and AWS KMS key policies](#) before you choose your just-in-time solution.

Temporary elevated access capabilities assessed for AWS partner validation

AWS Identity has validated that the temporary elevated access capabilities offered by [CyberArk Secure Cloud Access](#), [Tenable](#), and [Okta Access Requests](#) address the following common customer requirements:

- Users can request access to a permission set for a user-specified time period, specifying the AWS account, permission set, time period, and reason.
- Users can receive approval status for their request.
- Users can't invoke a session with a given scope, unless there is an approved request with the same scope and they invoke the session during the approved time period.
- There is a way to specify who can approve requests.
- Approvers can't approve their own requests.
- Approvers have a list of pending, approved, and rejected requests and can export it for auditors.
- Approvers can approve and reject pending requests.
- Approvers can add a note explaining their decision.
- Approvers can revoke an approved request, preventing future use of elevated access.

Note

If a user is signed in with elevated access when an approved request is revoked, their session remains active for up to one hour after the approval is revoked. For information about authentication sessions, see [Authentication in IAM Identity Center](#).

- User actions and approvals are available for audit.

Single sign-on access to AWS accounts

You can assign users in your connected directory permissions to the management account or member accounts in your organization in AWS Organizations based on [common job functions](#). Or you can use custom permissions to meet your specific security requirements. For example, you can grant database administrators broad permissions to Amazon RDS in development accounts but limit their permissions in production accounts. IAM Identity Center configures all the necessary user permissions in your AWS accounts automatically.

Note

You might need to grant users or groups permissions to operate in the AWS Organizations management account. Because it is a highly privileged account, additional security restrictions require you to have the [IAMFullAccess](#) policy or equivalent permissions before you can set this up. These additional security restrictions are not required for any of the member accounts in your AWS organization.

Topics

- [Assign user access to AWS accounts](#)
- [Remove user and group access to an AWS account](#)
- [Revoke active IAM role sessions created by permission sets](#)
- [Delegate who can assign single sign-on access to users and groups in the management account](#)

Assign user access to AWS accounts

Use the following procedure to assign single sign-on access to users and groups in your connected directory and use permission sets to determine their level of access.

To check existing user and group access, see [View user and group assignments](#).

Note

To simplify administration of access permissions, we recommended that you assign access directly to groups rather than to individual users. With groups you can grant or deny

permissions to groups of users rather than having to apply those permissions to each individual. If a user moves to a different organization, you simply move that user to a different group and they automatically receive the permissions that are needed for the new organization.

To assign user or group access to AWS accounts

1. Open the [IAM Identity Center console](#).

Note

Make sure that the IAM Identity Center console is using the Region where your AWS Managed Microsoft AD directory is located before you move to the next step.

2. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
3. On the **AWS accounts** page, a tree view list of your organization appears. Select the check box next to one or more AWS accounts to which you want to assign single sign-on access.

Note

You can select up to 10 AWS accounts at a time per permission set when you assign single sign-on access to users and groups. To assign more than 10 AWS accounts to the same set of users and groups, repeat this procedure as required for the additional accounts. When prompted, select the same users, groups, and permission set.

4. Choose **Assign users or groups**.
5. For **Step 1: Select users and groups**, on the **Assign users and groups to "AWS-account-name"** page, do the following:
 1. On the **Users** tab, select one or more users to whom to grant single sign-on access.

To filter the results, start typing the name of the user that you want in the search box.
 2. On the **Groups** tab, select one or more groups to which to grant single sign-on access.

To filter the results, start typing the name of the group that you want in the search box.
 3. To display the users and groups that you selected, choose the sideways triangle next to **Selected users and groups**.

4. After you confirm that the correct users and groups are selected, choose **Next**.
6. For **Step 2: Select permission sets**, on the **Assign permission sets to "AWS-account-name"** page, do the following:
 1. Select one or more permission sets. If required, you can create and select new permission sets.
 - To select one or more existing permission sets, under **Permission sets**, select the permission sets that you want to apply to the users and groups that you selected in the previous step.
 - To create one or more new permission sets, choose **Create permission set**, and follow the steps in [Create a permission set](#). After you create the permission sets that you want to apply, in the IAM Identity Center console, return to **AWS accounts** and follow the instructions until you reach **Step 2: Select permission sets**. When you reach this step, select the new permission sets that you created, and proceed to the next step in this procedure.
 2. After you confirm that the correct permission sets are selected, choose **Next**.
7. For **Step 3: Review and Submit**, on the **Review and submit assignments to "AWS-account-name"** page, do the following:
 1. Review the selected users, groups, and permission sets.
 2. After you confirm that the correct users, groups, and permission sets are selected, choose **Submit**.

 **Important**

The user and group assignment process might take a few minutes to complete. Leave this page open until the process successfully completes.

 **Note**

You might need to grant users or groups permissions to operate in the AWS Organizations management account. Because it is a highly privileged account, additional security restrictions require you to have the [IAMFullAccess](#) policy or equivalent permissions before you can set this up. These additional security

restrictions are not required for any of the member accounts in your AWS organization.

Remove user and group access to an AWS account

Use this procedure to remove single sign-on access to an AWS account for one or more users and groups in your connected directory.

To remove user and group access to an AWS account

1. Open the [IAM Identity Center console](#).
2. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
3. On the **AWS accounts** page, a tree view list of your organization appears. Select the name of the AWS account that contains the users and groups for whom you want to remove single sign-on access.
4. On the **Overview** page for the AWS account, under **Assigned users and groups**, select the name of one or more users or groups, and choose **Remove access**.
5. In the **Remove access** dialog box, confirm that the names of the users or groups are correct, and choose **Remove access**.

Revoke active IAM role sessions created by permission sets

The following is a general procedure for revoking an active permission set session for an IAM Identity Center user. The procedure assumes that you want to remove all access for a user who has compromised credentials or for a bad actor who is in the system. The prerequisite is to have followed the guidance in [Prepare to revoke an active IAM role session created by a permission set](#). We assume that the deny all policy is present in a service control policy (SCP).

Note

AWS recommends you build automation to handle all steps except console-only operations.

1. **Obtain the user ID of the person whose access you must revoke.** You can use the identity store APIs to find the user by their username.

2. **Update the Deny policy to add the user ID from step 1 in your service control policy (SCP).** After completing this step, the target user loses access and is unable to take actions with any roles that the policy affects.
3. **Remove all permission set assignments for the user.** If access is assigned through group memberships, remove the user from all groups and all direct permission set assignments. This step prevents the user from assuming any additional IAM roles. If a user has an active AWS access portal session and you disable the user, they can continue to assume new roles until you remove their access.
4. **If you use an identity provider (IdP) or Microsoft Active Directory as an identity source, disable the user in the identity source.** Disabling the user prevents the creation of additional AWS access portal sessions. Use your IdP or Microsoft Active Directory API documentation to learn how to automate this step. If you're using the IAM Identity Center directory as an identity source, don't disable user access yet. You'll disable user access in step 6.
5. **In the IAM Identity Center console, find the user and delete their active session.**
 - a. Choose **Users**.
 - b. Choose the user whose active session you want to delete.
 - c. On the user's detail page, choose the **Active sessions** tab.
 - d. Select the check boxes next to the sessions you want to delete and choose **Delete session**.

This ensures the user's AWS access portal session stops within approximately 60 minutes. Learn about [session duration](#).

6. **In the IAM Identity Center console, disable user access.**
 - a. Choose **Users**.
 - b. Choose the user whose access you want to disable.
 - c. On the user's detail page, expand **General information** and choose the **Disable user access** button to prevent further logins of the user.
7. **Leave the Deny policy in place for at least 12 hours.** Otherwise, the user with an active IAM role session will have restored actions with the IAM role. If you wait 12 hours, active sessions expire and the user won't be able to access the IAM role again.

⚠ Important

If you disable a user's access before stopping the user session (you completed step 6 without completing step 5), you can no longer stop the user session through the IAM Identity Center console. If you inadvertently disable user access before stopping the user session, you can re-enable the user, stop their session, and then disable their access again.

You can now change the user's credentials if their password was compromised and [restore their assignments](#).

Delegate who can assign single sign-on access to users and groups in the management account

Assigning single sign-on access to the management account using the IAM Identity Center console is a privileged action. By default, only an AWS account root user or a user who has the **AWSSSOMasterAccountAdministrator** and **IAMFullAccess** AWS managed policies attached, can assign single sign-on access to the management account. The **AWSSSOMasterAccountAdministrator** and **IAMFullAccess** policies manage single sign-on access to the management account within an AWS Organizations organization.

Use the following steps to delegate permissions to manage single sign-on access to users and groups in your directory.

To grant permissions to manage single sign-on access to users and groups in your directory

1. Sign in to the IAM Identity Center console as a root user of the management account or with another user who has administrator permissions to the management account.
2. Follow the steps in [Create a permission set](#) to create a permission set, and then do the following:
 1. On the **Create new permission set** page, select the **Create a custom permission set** check box, and then choose **Next: Details**.
 2. On the **Create new permission set page**, specify a name for the custom permission set and optionally, a description. If required, modify the session duration and specify a relay state URL.

Note

For the relay state URL, you must specify a URL that is in the AWS Management Console. For example:

`https://console.aws.amazon.com/ec2/`

For more information, see [Set relay state for quick access to the AWS Management Console](#).

3. Under **What policies do you want to include in your permission set?**, select the **Attach AWS managed policies** check box.
 4. In the list of IAM policies, choose both the **AWSSOMasterAccountAdministrator** and **IAMFullAccess** AWS managed policies. These policies grant permissions to any user and groups who are assigned access to this permission set in the future.
 5. Choose **Next: Tags**.
 6. Under **Add tags (optional)**, specify values for **Key** and **Value (optional)**, and then choose **Next: Review**. For more information about tags, see [Tagging AWS IAM Identity Center resources](#).
 7. Review the selections you made, and then choose **Create**.
3. Follow the steps in [Assign user access to AWS accounts](#) to assign the appropriate users and groups to the permission set that you just created.
 4. Communicate the following to the assigned users: When they sign in to the AWS access portal and choose the **Accounts** tab, they must choose the appropriate role name to be authenticated with the permissions that you just delegated.

Manage AWS accounts with permission sets

A permission set is a template that you create and maintain that defines a collection of one or more [IAM policies](#). Permission sets simplify the assignment of AWS account access for users and groups in your organization. For example, you can create a *Database Admin* permission set that includes policies for administering AWS RDS, DynamoDB, and Aurora services, and use that single permission set to grant access to a list of target AWS accounts within your [AWS Organization](#) for your database administrators.

IAM Identity Center assigns access to a user or group in one or more AWS accounts with permission sets. When you assign a permission set, IAM Identity Center creates corresponding IAM Identity

Center-controlled IAM roles in each account, and attaches the policies specified in the permission set to those roles. IAM Identity Center manages the role, and allows the authorized users you've defined to assume the role, by using the IAM Identity Center User Portal or AWS CLI. As you modify the permission set, IAM Identity Center ensures that the corresponding IAM policies and roles are updated accordingly.

You can add [AWS managed policies](#), [customer managed policies](#), inline policies, and [AWS managed policies for job functions](#) to your permission sets. You can also assign an AWS managed policy or a customer managed policy as a [permissions boundary](#).

To create a permission set, see [Create, manage, and delete permission sets](#).

Topics

- [Predefined permissions for AWS managed policies](#)
- [Custom permissions for AWS managed and customer managed policies](#)
- [Create, manage, and delete permission sets](#)
- [Configure permission set properties](#)

Predefined permissions for AWS managed policies

You can create a predefined permission set with AWS managed policies.

When you create a permission set with predefined permissions, you choose one policy from a list of AWS managed policies. Within the available policies, you can choose from **Common permission policies** and **Job function policies**.

Common permission policies

Choose from a list of AWS managed policies that make it possible to access resources in your entire AWS account. You can add one of the following policies:

- AdministratorAccess
- PowerUserAccess
- ReadOnlyAccess
- ViewOnlyAccess

Job function policies

Choose from a list of AWS managed policies that make it possible to access resources in your AWS account that might be relevant to a job within your organization. You can add one of the following policies:

- Billing
- DataScientist
- DatabaseAdministrator
- NetworkAdministrator
- SecurityAudit
- SupportUser
- SystemAdministrator

For detailed descriptions of the available common permission policies and job function policies, see [AWS managed policies for job functions](#) in the *AWS Identity and Access Management user guide*.

For instructions on how to create a permission set, see [Create, manage, and delete permission sets](#).

Custom permissions for AWS managed and customer managed policies

You can create a permission set with **Custom permissions**, combining any of the AWS managed and customer managed policies that you have in AWS Identity and Access Management (IAM) along with inline policies. You can also include permissions boundary, setting the maximum possible permissions that other policies can grant to users of your permission set.

For instructions on how to create a permission set, see [Create, manage, and delete permission sets](#).

Policy types that you can attach to your permission set

Topics

- [Inline policies](#)
- [AWS managed policies](#)
- [Customer managed policies](#)
- [Permissions boundaries](#)

Inline policies

You can attach an *inline policy* to a permission set. An inline policy is a block of text formatted as an IAM policy that you add directly to your permission set. You can paste in a policy, or generate a new one with the policy creation tool in the IAM Identity Center console when you create a new permission set. You can also create IAM policies with the [AWS Policy Generator](#).

When you deploy a permission set with an inline policy, IAM Identity Center creates an IAM policy in the AWS accounts where you assign your permission set. IAM Identity Center creates the policy when you assign the permission set to the account. The policy is then attached to the IAM role in your AWS account that your user assumes.

When you create an inline policy and assign your permission set, IAM Identity Center configures the policies in your AWS accounts for you. When you build your permission set with [Customer managed policies](#), you must create the policies in your AWS accounts yourself before you assign the permission set.

AWS managed policies

You can attach *AWS managed policies* to your permission set. AWS managed policies are IAM policies that AWS maintains. In contrast, [Customer managed policies](#) are IAM policies in your account that you create and maintain. AWS managed policies address common least privilege use cases in your AWS account. You can assign an AWS managed policy as permissions for the role that IAM Identity Center creates, or as a [permissions boundary](#).

AWS maintains [AWS managed policies for job functions](#) that assign job-specific access permissions to your AWS resources. You can add one job-function policy when you choose to use **Predefined permissions** with your permission set. When you choose **Custom permissions**, you can add more than one job-function policy.

Your AWS account also contains a large number of AWS managed IAM policies for specific AWS services and combinations of AWS services. When you create a permission set with **Custom permissions**, you can choose from many additional AWS managed policies to assign to your permission set.

AWS populates every AWS account with AWS managed policies. To deploy a permission set with AWS managed policies, you don't need to first create a policy in your AWS accounts. When you build your permission set with [Customer managed policies](#), you must create the policies in your AWS accounts yourself before you assign the permission set.

For more information about AWS managed policies, see [AWS managed policies](#) in the IAM User Guide.

Customer managed policies

You can attach *customer managed policies* to your permission set. Customer managed policies are IAM policies in your account that you create and maintain. In contrast, [AWS managed policies](#) are IAM policies in your account that AWS maintains. You can assign a customer managed policy as permissions for the role that IAM Identity Center creates, or as a [permissions boundary](#).

When you create a permission set with a customer managed policy, you must create an IAM policy with the same name and path in each AWS account where IAM Identity Center assigns your permission set. If you are specifying a custom path, make sure to specify the same path in each AWS account. For more information, see [Friendly names and paths](#) in the *IAM User Guide*. IAM Identity Center attaches the IAM policy to the IAM role that it creates in your AWS account. As a best practice, apply the same permissions to the policy in each account where you assign the permission set. For more information, see [Use IAM policies in permission sets](#).

For more information, see [Customer managed policies](#) in the IAM User Guide.

Permissions boundaries

You can attach a *permissions boundary* to your permission set. A permissions boundary is an AWS managed or customer managed IAM policy that sets the maximum permissions that an identity-based policy can grant to an IAM principal. When you apply a permissions boundary, your [Inline policies](#), [Customer managed policies](#), and [AWS managed policies](#) can't grant any permissions that exceed the permissions that your permissions boundary grants. A permissions boundary doesn't grant any permissions, but instead makes it so that IAM ignores all permissions beyond the boundary.

When you create a permission set with a customer managed policy as a permissions boundary, you must create an IAM policy with the same name in each AWS account where IAM Identity Center assigns your permission set. IAM Identity Center attaches the IAM policy as a permissions boundary to the IAM role that it creates in your AWS account .

For more information, see [Permissions boundaries for IAM entities](#) in the IAM User Guide.

Create, manage, and delete permission sets

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in IAM Identity Center and can be provisioned to one or more AWS accounts. You can assign more than one permission set to a user. For more information about permission sets and how they are used in IAM Identity Center, see [Manage AWS accounts with permission sets](#).

Keep the following considerations in mind when creating permissions sets:

- **Start with a predefined permission set**

With a predefined permission set, which uses [predefined permissions](#), you choose a single AWS managed policy from a list of available policies. Each policy grants a specific level of access to AWS services and resources or permissions for a common job function. For information about each of these policies, see [AWS managed policies for job functions](#). After you have collected usage data you can refine the permission set to be more restrictive.

- **Limit management session duration to reasonable work periods**

When users federate into their AWS account and use the AWS Management Console or the AWS Command Line Interface (AWS CLI), IAM Identity Center uses the session duration setting on the permission set to control the duration of the session. When the user session reaches the session duration they're signed out of the console and asked to sign in again. As a security best practice, we recommend that you don't set the session duration length longer than is needed to perform the role. By default, the value for **Session duration** is one hour. You can specify a maximum value of 12 hours. For more information, see [Set session duration for AWS accounts](#).

- **Limit workforce user portal session duration**

Workforce users use portal sessions to choose roles and access applications. By default, the value for **Maximum session duration**, which determines the length of time that a workforce user can be signed in to the AWS access portal before they must re-authenticate, is eight hours. You can specify a maximum value of 90 days. For more information, see [Configure the session duration of the AWS access portal and IAM Identity Center integrated applications](#).

- **Use the role that provides least-privilege permissions**

Each permission set that you create and assign to your user appears as an available role in the AWS access portal. When you sign in to the portal as that user, choose the role that corresponds to the most restrictive permission set that you can use to perform tasks in the account, rather

than AdministratorAccess. Test your permission sets to verify they provide the necessary access before sending the user invitation.

Note

You can also use [AWS CloudFormation](#) to create and assign permission sets and assign users to those permission sets.

Topics

- [Create a permission set](#)
- [View and change a permission set](#)
- [Delegate permission set administration](#)
- [Use IAM policies in permission sets](#)
- [Delete permission sets](#)

Create a permission set

Use this procedure to create a predefined permission set that uses a single AWS managed policy, or a custom permission set that uses up to 10 AWS managed or customer managed policies and an inline policy. You can request an adjustment to the maximum number of 10 policies in the [Service Quotas console](#) for IAM.

You can create a permission set in the IAM Identity Center console.

To create a permission set

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Choose **Create permission set**.
4. On the **Select permission set type** page, under **Permission set type**, select a permission set type.
5. Choose one or more policies that you want to use for the permission set, based on the permission set type:
 - **Predefined permission set**

1. Under **Policy for predefined permission set**, select one of the IAM **Job function policies** or **Common permission policies** in the list, and then choose **Next**. For more information, see [AWS managed policies for job functions](#) and [AWS managed policies](#) in the *AWS Identity and Access Management User Guide*.
 2. Go to Step 6 to complete the **Specify permission set details** page.
- **Custom permission set**
 1. Choose **Next**.
 2. On the **Specify policies and permission boundary** page, choose the types of IAM policies that you want to apply to your new permission set. By default, you can add any combination of up to 10 **AWS managed policies** and **Customer managed policies** to your permission set. This quota is set by IAM. To raise it, request an increase to the IAM quota *Managed policies attached to an IAM role* in the Service Quotas console in each AWS account where you want to assign the permission set.
 - Expand **AWS managed policies** to add policies from IAM that AWS builds and maintains. For more information, see [AWS managed policies](#).
 - a. Search for and choose **AWS managed policies** that you want to apply to your users in the permission set.
 - b. If you want to add another type of policy, choose its container and make your selection. Choose **Next** when you've chosen all the policies that you want to apply. Go to Step 6 to complete the **Specify permission set details** page.
 - Expand **Customer managed policies** to add policies from IAM that you build and maintain. For more information, see [Customer managed policies](#).
 - a. Choose **Attach policies** and enter the name of a policy that you want to add to your permission set. In each account where you want to assign the permission set, create a policy with the name you entered. As a best practice, assign the same permissions to the policy in each account.
 - b. Choose **Attach more** to add another policy.
 - c. If you want to add another type of policy, choose its container and make your selection. Choose **Next** when you've chosen all the policies that you want to apply. Go to Step 6 to complete the **Specify permission set details** page.
 - Expand **Inline policy** to add custom JSON-formatted policy text. Inline policies don't correspond to existing IAM resources. To create an inline policy, enter custom policy language in the provided form. IAM Identity Center adds the policy to the IAM resources that it creates in your member accounts. For more information, see [Inline policies](#).

- a. Add your desired actions and resources within the interactive editor to your inline policy. Additional statements can be added with **Add new statement**.
 - b. If you want to add another type of policy, choose its container and make your selection. Choose **Next** when you've chosen all the policies that you want to apply. Go to Step 6 to complete the **Specify permission set details** page.
 - Expand **Permissions boundary** to add an AWS managed or customer managed IAM policy as the maximum permissions that your other policies in the permission set can assign. For more information, see [Permissions boundaries](#).
 - a. Choose **Use a permissions boundary to control the maximum permissions**.
 - b. Choose **AWS managed policy** to set a policy from IAM that AWS builds and maintains as your permissions boundary. Chose **Customer managed policies** to set a policy from IAM that *you* build and maintain as your permissions boundary.
 - c. If you want to add another type of policy, choose its container and make your selection. Choose **Next** when you've chosen all the policies that you want to apply. Go to Step 6 to complete the **Specify permission set details** page.
6. On the **Specify permission set details** page, do the following:
1. Under **Permission set name**, type a name to identify this permission set in IAM Identity Center. The name that you specify for this permission set appears in the AWS access portal as an available role. Users sign into the AWS access portal, choose an AWS account, and then choose the role.
 2. (Optional) You can also type a description. The description appears in the IAM Identity Center console only, not the AWS access portal.
 3. (Optional) Specify the value for **Session duration**. This value determines the length of time that a user can be logged on before the console logs them out of their session. For more information, see [Set session duration for AWS accounts](#).
 4. (Optional) Specify the value for **Relay state**. This value is used in the federation process to redirect users within the account. For more information, see [Set relay state for quick access to the AWS Management Console](#).

 **Note**

The relay state URL must be within the AWS Management Console. For example:
`https://console.aws.amazon.com/ec2/`

5. Expand **Tags (optional)**, choose **Add tag**, and then specify values for **Key** and **Value (optional)**.

For information about tags, see [Tagging AWS IAM Identity Center resources](#).

6. Choose **Next**.
7. On the **Review and create** page, review the selections that you made, and then choose **Create**.
8. By default, when you create a permission set, the permission set isn't provisioned (used in any AWS accounts). To provision a permission set in an AWS account, you must assign IAM Identity Center access to users and groups in the account, and then apply the permission set to those users and groups. For more information, see [Single sign-on access to AWS accounts](#).

View and change a permission set

You can view and change a permission set with the AWS IAM Identity Center console. For more information about permission sets and how they are used in IAM Identity Center, see [the section called "Permission sets"](#).

View a permission set

Use this procedure to view a permission set with the AWS IAM Identity Center console.

1. Sign in to the AWS Management Console and open the AWS IAM Identity Center console at <https://console.aws.amazon.com/singlesignon/>.
2. Under **Multi-account permissions**, choose **Permission sets**.
3. On the **Permission sets** page, select the permission set you want to view.
4. On the selected permission sets page, you can [delete](#), edit, and attach policies to the permission set.

Change a permission set

Use this procedure to change a permission set with the AWS IAM Identity Center console.

1. Sign in to the AWS Management Console and open the AWS IAM Identity Center console at <https://console.aws.amazon.com/singlesignon/>.
2. Under **Multi-account permissions**, choose **AWS accounts**.
3. On the **AWS account** page, a tree view list of your organization appears. Select the name of the AWS account from which you want to change the permission set.

4. On the **Overview** page of the AWS account, under **Assigned Users and Groups**, select the username or group name of the permission set you want to change. Then choose **Change permission sets**.
5. Make the desired changes to the permission set and then choose **Save changes**.
6. Navigate to the **Permission sets** tab and select the recently changed permission set and choose **Update**.
7. On the **Update permissions** page, choose **Update**.

Delegate permission set administration

IAM Identity Center enables you to delegate management of permission sets and assignments in accounts by creating [IAM policies](#) that reference the [Amazon Resource Names \(ARNs\)](#) of IAM Identity Center resources. For example, you can create policies that enable different administrators to manage assignments in specified accounts for permission sets with specific tags.

You can use either of the following methods to create these types of policies.

- (Recommended) Create [permission sets](#) in IAM Identity Center, each with a different policy, and assign the permission sets to different users or groups. This enables you to manage administrative permissions for users who sign in using your chosen [IAM Identity Center identity source](#).
- Create custom policies in IAM, and then attach them to IAM roles that your administrators assume. For information about roles, see [IAM roles](#) to get their assigned IAM Identity Center administrative permissions.

Important

IAM Identity Center resource ARNs are case sensitive.

The following shows the proper case for referencing the IAM Identity Center permission set and account resource types.

Resource Types	ARN	Context Keys
PermissionSet	arn:\${Partition}:sso::permissionSet/\${InstanceId}/\${PermissionSetId}	aws:ResourceTag/\${TagKey}
Account	arn:\${Partition}:sso::account/\${AccountId}	Not Applicable

Use IAM policies in permission sets

In [Create a permission set](#), you learned how to add policies, including customer managed policies and permissions boundaries, to a permission set. When you add customer managed policies and permissions to a permission set, IAM Identity Center doesn't create a policy in any AWS accounts. You must instead create those policies in advance in each account where you want to assign your permission set, and match them to the name and path specifications of your permission set. When you assign a permission set to an AWS account in your organization, IAM Identity Center creates an [AWS Identity and Access Management \(IAM\) role](#) and attaches your [IAM policies](#) to that role.

Note

Before you assign your permission set with IAM policies, you must prepare your member account. The name of an IAM policy in your member account must be a match to the name of the policy in your management account. IAM Identity Center fails to assign the permission set if the policy doesn't exist in your member account. The permissions that the policy grants don't have to be an exact match between accounts.

Assign an IAM policy to a permission set

1. Create an IAM policy in each of the AWS accounts where you want to assign the permission set.
2. Assign permissions to the IAM policy. You can assign different permissions in different accounts. For a consistent experience, configure and maintain identical permissions in each policy. You can use automation resources like AWS CloudFormation StackSets to create copies of an IAM policy with the same name and permissions in each member account. For

more information about CloudFormation StackSets, see [Working with AWS CloudFormation StackSets](#) in the *AWS CloudFormation User guide*.

3. Create a permission set in your management account and add your IAM policy under **Customer managed policies** or **Permissions boundary**. For more details about how to create a permission set, See [Create a permission set](#).
4. Add any inline policies, AWS managed policies, or additional IAM policies that you have prepared.
5. Create and assign your permission set.

Delete permission sets

If you want to revoke an active permission set session, see [Revoke active IAM role sessions created by permission sets](#).

Before you can delete a permission set from IAM Identity Center, you must remove it from all AWS accounts that use the permission set. To check existing user and group access, see [View user and group assignments](#).

To remove a permission set from an AWS account

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **AWS accounts**.
3. On the **AWS accounts** page, a tree view list of your organization appears. Select the name of the AWS account from which you want to remove the permission set.
4. On the **Overview** page for the AWS account, choose the **Permission sets** tab.
5. Select the check box next to the permission set that you want to remove, and then choose **Remove**.
6. In the **Remove permission set** dialog box, confirm that the correct permission set is selected, type **Delete** to confirm removal, and then choose **Remove access**.

Use the following procedure to delete one or more permission sets so that they can no longer be used by any AWS account in the organization.

Note

All users and groups that have been assigned this permission set, regardless of what AWS account is using it, will no longer be able to sign in. To check existing user and group access, see [View user and group assignments](#).

To delete a permission set from an AWS account

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Select the permission set that you want to delete, and then choose **Delete**.
4. In the **Delete permission set** dialog box, type the name of the permission set to confirm deletion, and then choose **Delete**. The name is case-sensitive.

Configure permission set properties

In IAM Identity Center you can customize the user experience by configuring the following permission set properties.

Topics

- [Set session duration for AWS accounts](#)
- [Set relay state for quick access to the AWS Management Console](#)
- [Use a Deny policy to revoke active user permissions](#)

Set session duration for AWS accounts

For each [permission set](#), you can specify a session duration to control the length of time that a user can be signed in to an AWS account. When the specified duration elapses, AWS signs the user out of the session.

When you create a new permission set, the session duration is set to 1 hour (in seconds) by default. The minimum session duration is 1 hour, and can be set to a maximum of 12 hours. IAM Identity Center automatically creates IAM roles in each assigned account for each permission set, and configures these roles with a maximum session duration of 12 hours.

When users federate into their AWS account console or when the AWS Command Line Interface (AWS CLI) is used, IAM Identity Center uses the session duration setting on the permission set to control the duration of the session. By default, IAM roles generated by IAM Identity Center for permission sets can only be assumed by IAM Identity Center users, which ensures that the session duration specified in the IAM Identity Center permission set is enforced.

Important

As a security best practice, we recommend that you do not set the session duration length longer than is needed to perform the role.

After you create a permission set, you can update it to apply a new session duration. Use the following procedure to modify the session duration length for a permission set.

To set the session duration

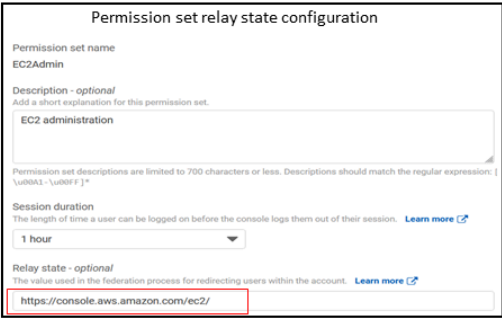
1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Choose the name of the permission set for which you want to change the session duration.
4. On the details page for the permission set, to the right of the **General settings** section heading, choose **Edit**.
5. On the **Edit general permission set settings** page, choose a new value for **Session duration**.
6. If the permission set is provisioned in any AWS accounts, the names of the accounts appear under **AWS accounts to reprovision automatically**. After the session duration value for the permission set is updated, all AWS accounts that use the permission set are reprovisioned. This means that the new value for this setting is applied to all AWS accounts that use the permission set.
7. Choose **Save changes**.
8. At the top of the **AWS accounts** page, a notification appears.
 - If the permission set is provisioned in one or more AWS accounts, the notification confirms that the AWS accounts were reprovisioned successfully, and the updated permission set was applied to the accounts.
 - If the permission set isn't provisioned in an AWS account, the notification confirms that the settings for the permission set were updated.

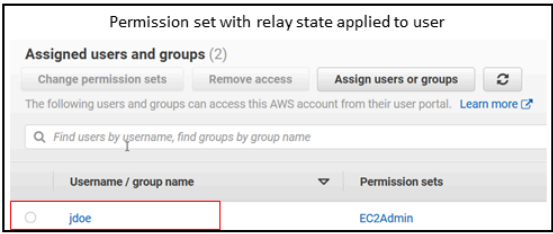
Set relay state for quick access to the AWS Management Console

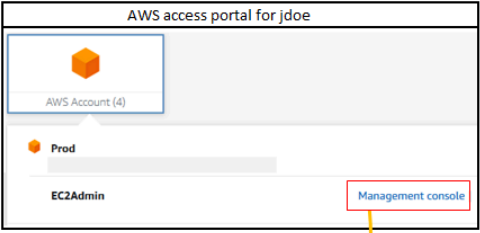
By default, when a user signs into the AWS access portal, chooses an account, and then chooses the role that AWS creates from the assigned permission set, IAM Identity Center redirects the user's browser to the AWS Management Console. You can change this behavior by setting the relay state to a different console URL.

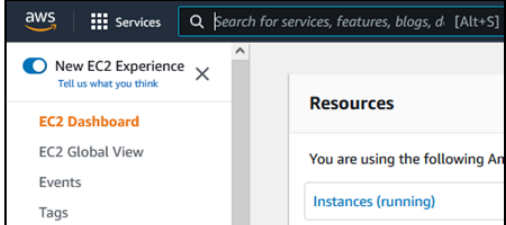
Setting the relay state enables you to provide the user with quick access to the console that's most appropriate for their role. For example, you can set the relay state to the Amazon EC2 console URL (<https://console.aws.amazon.com/ec2/>) to redirect the user to that console when they choose the Amazon EC2 administrator role. During the redirection to the default URL or relay state URL, IAM Identity Center routes the user's browser to the console endpoint in the last AWS Region used by the user. For example, if a user ended their last console session in the Europe (Stockholm) Region (eu-north-1), the user is redirected to the Amazon EC2 console in that Region.

- Administrator for AWS IAM Identity Center (successor to AWS Single Sign-On) sets the relay state


- IAM Identity Center administrator assigns single sign-on access to user and applies permission set with relay state


- User signs in and chooses Management console


- IAM Identity Center redirects user to the Amazon EC2 console in the user's last used Region



A yellow arrow points from the 'Management console' button in step 3 to the 'EC2 Dashboard' in step 4, indicating the redirection.

To configure IAM Identity Center to redirect the user to a console in a specific AWS Region, include the Region specification as part of the URL. For example, to redirect the user to the Amazon EC2 console in the US East (Ohio) Region (us-east-2), specify the URL for the Amazon EC2 console in that Region (<https://us-east-2.console.aws.amazon.com/ec2/>). If you enabled IAM

Identity Center in the US West (Oregon) Region (us-west-2) Region and you want to direct the user to that Region, specify `https://us-west-2.console.aws.amazon.com`.

Configure the relay state

Use the following procedure to configure the relay state URL for a permission set.

1. Open the [IAM Identity Center console](#).
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Choose the name of the permission set for which you want to set the new relay state URL.
4. On the details page for the permission set, to the right of the **General settings** section heading, choose **Edit**.
5. On the **Edit general permission set settings** page, under **Relay state**, type a console URL for any of the AWS services. For example:

`https://console.aws.amazon.com/ec2/`

Note

The relay state URL must be within the AWS Management Console.

6. If the permission set is provisioned in any AWS accounts, the names of the accounts appear under **AWS accounts to reprovision automatically**. After the relay state URL for the permission set is updated, all AWS accounts that use the permission set are reprovisioned. This means that the new value for this setting is applied to all AWS accounts that use the permission set.
7. Choose **Save changes**.
8. At the top of the **AWS Organization** page, a notification appears.
 - If the permission set is provisioned in one or more AWS accounts, the notification confirms that the AWS accounts were reprovisioned successfully, and the updated permission set was applied to the accounts.
 - If the permission set isn't provisioned in an AWS account, the notification confirms that the settings for the permission set were updated.

Note

You can automate this process by using the AWS API, an AWS SDK, or the AWS Command Line Interface(AWS CLI). For more information, see:

- The `CreatePermissionSet` or `UpdatePermissionSet` actions in the [IAM Identity Center API Reference](#)
- The `create-permission-set` or `update-permission-set` commands in the [sso-admin](#) section of the *AWS CLI Command Reference*.

Use a Deny policy to revoke active user permissions

You might need to revoke an IAM Identity Center user's access to AWS accounts while the user is actively using a permission set. You can remove their ability to use their active IAM role sessions by implementing a Deny policy for an unspecified user in advance, then when needed, you can update the Deny policy to specify the user whose access you want to block. This topic explains how to create a Deny policy and considerations for how to deploy the policy.

Prepare to revoke an active IAM role session created by a permission set

You can prevent the user from taking actions with an IAM role they're actively using by applying a deny all policy for a specific user through the use of a Service Control Policy. You can also prevent a user from using any permission set until you change their password, which removes a bad actor actively misusing stolen credentials. If you need to deny access broadly and prevent a user from re-entering a permission set or accessing other permission sets, you might also remove all user access, stop the active AWS access portal session, and disable the user sign-in. See [Revoke active IAM role sessions created by permission sets](#) to learn how to use the Deny policy in conjunction with additional actions for broader access revocation.

Deny policy

You can use a Deny policy with a condition that matches to the user's `UserID` from the IAM Identity Center identity store to prevent further actions by an IAM role that the user is actively using. Using this policy avoids impact to other users who might be using the same permission set when you deploy the Deny policy. This policy uses the placeholder user ID, *Add user ID here*, for `"identitystore:userId"` that you'll update with the user ID for which you want to revoke access.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "identitystore:userId": "Add user ID here"
        }
      }
    }
  ]
}
```

Although you could use another condition key such as `aws:userId`, `identitystore:userId` is certain because it's a globally unique value that's associated with one person. Using `aws:userId` in the condition can be affected by how user attributes are synchronized from your source of identities and can change if the user's username or email address changes.

From the IAM Identity Center console, you can find a user's `identitystore:userId` by navigating to **Users**, searching for the user by name, expanding the **General information** section and copying the User ID. It's also convenient to stop a user's AWS access portal session and disable their sign-in access in the same section while searching for the User ID. You can automate the process to create a Deny policy by obtaining the user's User ID through querying the identity store APIs.

Deploying the deny policy

You can use a placeholder user ID that isn't valid, such as *Add user ID here*, to deploy the Deny policy in advance using a Service Control Policy (SCP) that you attach to the AWS accounts users might have access to. This is the recommended approach for its ease and speed of impact. When you revoke a user's access with the Deny policy, you'll edit the policy to replace the placeholder user ID with the user ID of the person whose access you want to revoke. This prevents the user from taking any actions with any permission set in every account that you attach the SCP. It blocks the user's actions even if they use their active AWS access portal session to navigate to different

accounts and assume different roles. With the user's access fully blocked by the SCP, you can then disable their ability to sign in, revoke their assignments, and stop their AWS access portal session if needed.

As an alternative to using SCPs, you can also include the Deny policy in the inline policy of permission sets and in customer managed policies that are used by the permission sets the user can access.

If you must revoke access for more than one person, you can use a list of values in the condition block, such as:

```

"Condition": {
  "StringEquals": {
    "identitystore:userId": [" user1 userId", "user2 userId"...]
  }
}

```

Important

Regardless of the method(s) you use, you must take any other corrective actions and keep the user's user ID in the policy for at least 12 hours. After that time, any roles the user has assumed expire and you can then remove their user ID from the Deny policy.

Referencing permission sets in resource policies, Amazon EKS Cluster config maps, and AWS KMS key policies

When you assign a permission set to an AWS account, IAM Identity Center creates a role with a name that begins with `AWSReservedSSO_`.

The complete name and Amazon Resource Name (ARN) for the role use the following format:

Name	ARN
<code>AWSReservedSSO_ <i>permission-set-name</i> <i>e_unique-suffix</i></code>	<code>arn:aws:iam:: <i>aws-account-ID</i>:role/aws-reserved/sso.amazonaws.com/ <i>aws-region</i> /AWSReser</code>

Name	ARN
	vedSSO_ <i>permission-set-name_unique-suffix</i>

If your identity source in IAM Identity Center is hosted in us-east-1, there is no *aws-region* in the ARN. The complete name and ARN for the role use the following format:

Name	ARN
AWSReservedSSO_ <i>permission-set-name_unique-suffix</i>	arn:aws:iam:: <i>aws-account-ID</i> :role/ aws-reserved/sso.amazonaws.com/ AWSReservedSSO_ <i>permission-set-name_unique-suffix</i>

For example, if you create a permission set that grants AWS account access to database administrators, a corresponding role is created with the following name and ARN:

Name	ARN
AWSReservedSSO_DatabaseAdministrator_1234567890abcdef	arn:aws:iam::111122223333:role/ aws-reserved/sso.amazonaws.com/ eu-west-2/AWSReservedSSO_DatabaseAdministrator_1234567890abcdef

If you delete all assignments to this permission set in the AWS account, the corresponding role that IAM Identity Center created is also deleted. If you make a new assignment to the same permission set later, IAM Identity Center creates a new role for the permission set. The name and ARN of the new role include a different, unique suffix. In this example, the unique suffix is **abcdef0123456789**.

Name	ARN
AWSReservedSSO_DatabaseAdministrator_ abcdef0123456789	arn:aws:iam::111122223333:role/ aws-reserved/sso.amazonaws.com/

Name	ARN
	eu-west-2/AWSReservedSSO_DataAdministrator_ abcdef0123456789

The suffix change in the new name and ARN for the role will cause any policies that reference the original name and ARN to be out-of-date, which disrupts access for individuals who use the corresponding permission set. For example, a change in the ARN for the role will disrupt access for users of the permission set if the original ARN is referenced in the following configurations:

- In the `aws-auth ConfigMap` file for Amazon Elastic Kubernetes Service (Amazon EKS) clusters when you use the `aws-auth ConfigMap` for cluster access.
- In a resource-based policy for an AWS Key Management Service (AWS KMS) key. This policy is also referred to as a key policy.

Note

We recommend that you use [Amazon EKS access entries](#) to manage access to your Amazon EKS clusters. This allows you to use IAM permissions to manage the principals that have access to an Amazon EKS cluster. By using Amazon EKS access entries, you can use an IAM principal with Amazon EKS permissions to regain access to a cluster without contacting AWS Support.

Although you can update resource-based policies for most AWS services to reference a new ARN for a role that corresponds to a permission set, you must have a backup role that you create in IAM for Amazon EKS and AWS KMS if the ARN changes. For Amazon EKS, the backup IAM role must exist in the `aws-auth ConfigMap`. For AWS KMS, it must exist in your key policies. If you don't have a backup IAM role with permissions to update the `aws-auth ConfigMap` or the AWS KMS key policy, contact AWS Support to regain access to those resources.

Recommendations to avoid access disruptions

To avoid access disruptions due to changes in the ARN for a role that corresponds to a permission set, we recommend that you do the following.

- **Maintain at least one permission set assignment.**

Maintain this assignment in the AWS accounts that contain the roles that you reference in the `aws-auth ConfigMap` for Amazon EKS, key policies in AWS KMS, or resource-based policies for other AWS services.

For example, if you create an `EKSAccess` permission set and reference the corresponding role ARN from AWS account `111122223333`, then permanently assign an administrative group to the permission set in that account. Because the assignment is permanent, IAM Identity Center won't delete the corresponding role, which eliminates the renaming risk. The administrative group will always have access without the risk of privilege escalation.

- **For Amazon EKS clusters that use `aws-auth ConfigMap` and AWS KMS: Include a role created in IAM.**

If you reference role ARNs for permission sets in an `aws-auth ConfigMap` for Amazon EKS cluster or in key policies for AWS KMS keys, we recommend that you also include at least one role that you create in IAM. The role must allow you to access the Amazon EKS cluster or manage the AWS KMS key policy. The permission set must be able to assume this role. That way, if the role ARN for a permission set changes, you can update the reference to the ARN in the `aws-auth ConfigMap` or AWS KMS key policy. The next section provides an example of how you can create a trust policy for a role that's created in IAM. The role can be assumed only by an `AdministratorAccess` permission set.

Custom trust policy example

Following is an example of a custom trust policy that provides an `AdministratorAccess` permission set with access to a role that's created in IAM. The key elements of this policy include:

- The `Principal` element of this trust policy specifies an AWS account principal. In this policy, principals in the AWS account `111122223333` with `sts:AssumeRole` permissions can assume the role that's created in IAM.
- The `Condition` element of this trust policy specifies additional requirements for principals that can assume the role created in IAM. In this policy, the permission set with the following role ARN can assume the role.

```
arn:aws:iam::111122223333:role/aws-reserved/sso.amazonaws.com/eu-west-2/  
AWSReservedSSO_AdministratorAccess_*
```

Note

The `Condition` element includes the `ArnLike` condition operator and uses a wildcard at the end of the permission set role ARN, rather than a unique suffix. This means that the policy allows the permission set to assume the role created in IAM even if the role ARN for the permission set changes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::111122223333:role/aws-reserved/
sso.amazonaws.com/eu-west-2/AWSReservedSSO_AdministratorAccess_*"
        }
      }
    }
  ]
}
```

Including a role that you create in IAM in such a policy will provide you with emergency access to your Amazon EKS clusters, AWS KMS keys, or other AWS resources if a permission set or all assignments to the permission set are accidentally deleted and re-created.

Attribute-based access control

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. You can use IAM Identity Center to manage access to your AWS resources across multiple AWS accounts using user attributes that come from any IAM Identity Center identity source. In AWS, these attributes are called tags. Using user attributes as tags in AWS helps you

simplify the process of creating fine-grained permissions in AWS and ensures that your workforce gets access only to the AWS resources with matching tags.

For example, you can assign developers Bob and Sally, who are from two different teams, to the same permission set in IAM Identity Center and then select the team name attribute for access control. When Bob and Sally sign in to their AWS accounts, IAM Identity Center sends their team name attribute in the AWS session so Bob and Sally can access AWS project resources only if their team name attribute matches the team name tag on the project resource. If Bob moves to Sally's team in the future, you can modify his access by simply updating his team name attribute in the corporate directory. When Bob signs in next time, he will automatically get access to the project resources of his new team without requiring any permissions updates in AWS.

This approach also helps in reducing the number of distinct permissions you need to create and manage in IAM Identity Center as users associated with the same permission sets can now have unique permissions based on their attributes. You can use these user attributes in IAM Identity Center permission sets and resource-based policies to implement ABAC to AWS resources and simplify permissions management at scale.

Benefits

The following are additional benefits of using ABAC in IAM Identity Center.

- **ABAC requires fewer permission sets** – Because you don't have to create different policies for different job functions, you create fewer permission sets. This reduces your permissions management complexity.
- **Using ABAC, teams can change and grow quickly** – Permissions for new resources are automatically granted based on attributes when resources are appropriately tagged upon creation.
- **Use employee attributes from your corporate directory with ABAC** – You can use existing employee attributes from any identity source configured in IAM Identity Center to make access control decisions in AWS.
- **Track who is accessing resources** – Security administrators can easily determine the identity of a session by reviewing the user attributes in AWS CloudTrail to track user activity in AWS.

For information about how to configure ABAC using the IAM Identity Center console, see [Attributes for access control](#). For information about how to enable and configure ABAC using the IAM Identity

Center APIs, see [CreateInstanceAccessControlAttributeConfiguration](#) in the *IAM Identity Center API Reference Guide*.

Topics

- [Checklist: Configuring ABAC in AWS using IAM Identity Center](#)
- [Attributes for access control](#)

Checklist: Configuring ABAC in AWS using IAM Identity Center

This checklist includes the configuration tasks that are necessary to prepare your AWS resources and to set up IAM Identity Center for ABAC access. Complete the tasks in this checklist in order. When a reference link takes you to a topic, return back to this topic so that you can proceed with the remaining tasks in this checklist.

Step	Task	Reference
1	Review how to add tags to all your AWS resources. To implement ABAC in IAM Identity Center, you'll first need to add tags to all your AWS resources that you want to implement ABAC for.	<ul style="list-style-type: none"> • Tagging AWS resources
2	Review how to configure your identity source in IAM Identity Center with the associated user identities and attributes in your identity store. IAM Identity Center lets you use user attributes from any supported IAM Identity Center identity source for ABAC in AWS.	<ul style="list-style-type: none"> • Manage your identity source
3	Based on the following criteria, determine which attributes you want to use for making access control decisions in AWS and send them to IAM Identity Center.	<ul style="list-style-type: none"> • Getting started
	<ul style="list-style-type: none"> • If you are using an external identity provider (IdP), decide whether you want to use attributes passed from the IdP or select attributes from within IAM Identity Center. 	<ul style="list-style-type: none"> • Choosing attributes when using an external identity provider as your identity source

Step	Task	Reference
	<ul style="list-style-type: none"> If you choose to have your IdP send attributes, configure your IdP to transmit the attributes in SAML assertions. See the <code>Optional</code> sections in the tutorial for your specific IdP. 	<ul style="list-style-type: none"> Getting started tutorials for IAM Identity Center
	<ul style="list-style-type: none"> If you use an IdP as your identity source and choose to select attributes in IAM Identity Center, investigate how to configure SCIM so that the attribute values come from your IdP. If you cannot use SCIM with your IdP, add the users and their attributes using the IAM Identity Center console User page. 	<ul style="list-style-type: none"> Provisioning an external identity provider into IAM Identity Center using SCIM Supported external identity provider attributes
4	<p>Select the attributes to use for ABAC using the Attributes for access control page in the IAM Identity Center console. From this page you can select attributes for access control from the identity source that you configured in step 2. After your identities and their attributes are in IAM Identity Center, you must create key-value pairs (mappings) which will be passed to your AWS accounts for use in access control decisions.</p>	<ul style="list-style-type: none"> Choosing attributes when using IAM Identity Center as your identity source Choosing attributes when using AWS Managed Microsoft AD as your identity source Default mappings Enable and configure attributes for access control

Step	Task	Reference
5	Create custom permissions policies within your permission set and use access control attributes to create ABAC rules so that users can only access resources with matching tags. User attributes that you configured in step 4 are used as tags in AWS for access control decisions. You can refer to the access control attributes in the permissions policy using the <code>aws:PrincipalTag/key</code> condition.	<ul style="list-style-type: none">• Create permission policies for ABAC in IAM Identity Center
6	In your various AWS accounts, assign users to permissions sets you created in step 5. Doing so ensures that when they federate into their accounts and access AWS resources, they only get access based on matching tags.	<ul style="list-style-type: none">• Assign user access to AWS accounts

After you complete these steps, users who federate into an AWS account using single sign-on will get access to their AWS resources based on matching attributes.

Attributes for access control

Attributes for access control is the name of the page in the IAM Identity Center console where you select user attributes that you want to use in policies to control access to resources. You can assign users to workloads in AWS based on existing attributes in the users' identity source.

For example, suppose you want to assign access to S3 buckets based on department names. While on the **Attributes for access control** page, you select the **Department** user attribute for use with attribute-based access control (ABAC). In the IAM Identity Center permission set, you then write a policy that grants users access only when the **Department** attribute matches the department tag that you assigned to your S3 buckets. IAM Identity Center passes the user's department attribute to the account being accessed. The attribute is then used to determine access based on the policy. For more information about ABAC, see [Attribute-based access control](#).

Getting started

How you get started configuring attributes for access control depends on which identity source you are using. Regardless of the identity source you choose, after you have selected your attributes you

need to create or edit permission set policies. These policies must grant user identities access to AWS resources.

Choosing attributes when using IAM Identity Center as your identity source

When you configure IAM Identity Center as the identity source, you first add users and configure their attributes. Next, navigate to the **Attributes for access control** page and select the attributes you want to use in policies. Finally, navigate to the **AWS accounts** page to create or edit permission sets to use the attributes for ABAC.

Choosing attributes when using AWS Managed Microsoft AD as your identity source

When you configure IAM Identity Center with AWS Managed Microsoft AD as your identity source, you first map a set of attributes from Active Directory to user attributes in IAM Identity Center. Next, navigate to the **Attributes for access control** page. Then choose which attributes to use in your ABAC configuration based on the existing set of SSO attributes mapped from Active Directory. Finally, author ABAC rules using the access control attributes in permission sets to grant user identities access to AWS resources. For a list of the default mappings for user attributes in IAM Identity Center to the user attributes in your AWS Managed Microsoft AD directory, see [Default mappings](#).

Choosing attributes when using an external identity provider as your identity source

When you configure IAM Identity Center with an external identity provider (IdP) as your identity source, there are two ways to use attributes for ABAC.

- You can configure your IdP to send the attributes through SAML assertions. In this case, IAM Identity Center passes the attribute name and value from the IdP through for policy evaluation.

Note

Attributes in SAML assertions will not be visible to you on the **Attributes for access control** page. You will have to know these attributes in advance and add them to access control rules when you author policies. If you decide to trust your external IdPs for attributes, then these attributes will always be passed when users federate into AWS accounts. In scenarios where the same attributes are coming to IAM Identity Center through SAML and SCIM, the SAML attributes value take precedence in access control decisions.

- You can configure which attributes you use from the **Attributes for access control** page in the IAM Identity Center console. The attributes values that you choose here replace the values for any matching attributes that come from an IdP through an assertion. Depending on whether you are using SCIM, consider the following:
 - If using SCIM, the IdP automatically synchronizes the attribute values into IAM Identity Center. Additional attributes that are required for access control might not be present in the list of SCIM attributes. In that case, consider collaborating with the IT admin in your IdP to send such attributes to IAM Identity Center via SAML assertions using the required `https://aws.amazon.com/SAML/Attributes/AccessControl:` prefix. For information about how to configure user attributes for access control in your IdP to send through SAML assertions, see the [Getting started tutorials for IAM Identity Center](#) for your IdP.
 - If you are not using SCIM, you must manually add the users and set their attributes just as if you were using IAM Identity Center as an identity source. Next, navigate to the **Attributes for access control** page and choose the attributes you want to use in policies.

For a complete list of supported attributes for user attributes in IAM Identity Center to the user attributes in your external IdPs, see [Supported external identity provider attributes](#).

To get started with ABAC in IAM Identity Center, see the following topics.

Topics

- [Enable and configure attributes for access control](#)
- [Create permission policies for ABAC in IAM Identity Center](#)

Enable and configure attributes for access control

To use ABAC in all cases, you must first enable ABAC using the IAM Identity Center console or the IAM Identity Center API. If you choose to use IAM Identity Center to select attributes, you use the **Attributes for access control** page in the IAM Identity Center console or the IAM Identity Center API. If you use an external identity provider (IdP) as an identity source and choose to send attributes through the SAML assertions, you configure your IdP to pass the attributes. If a SAML assertion passes any of these attributes, IAM Identity Center will replace the attribute value with the value from the IAM Identity Center identity store. Only attributes configured in IAM Identity Center will be sent over for making access control decisions when users federate into their accounts.

Note

You cannot view attributes configured and sent by an external IdP from the **Attributes for access control** page in the IAM Identity Center console. If you are passing access control attributes in the SAML assertions from your external IdP, then those attributes are directly sent to the AWS account when users federate in. The attributes won't be available in IAM Identity Center for mapping.

Enable attributes for access control

Use the following procedure to enable the attributes for access (ABAC) control feature using the IAM Identity Center console.

Note

If you have existing permission sets and you plan to enable ABAC in your IAM Identity Center instance, additional security restrictions require you to first have the `iam:UpdateAssumeRolePolicy` policy. These additional security restrictions are not required if you do not have any permission sets created in your account.

To enable Attributes for access control

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**
3. On the **Settings** page, locate the **Attributes for access control** information box, and then choose **Enable**. Continue to the next procedure to configure it.

Select your attributes for access control

Use the following procedure to set up attributes for your ABAC configuration.

To select your attributes using the IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**

- On the **Settings** page, choose the **Attributes for access control** tab, and then choose **Manage attributes**.
- On the **Attributes for access control** page, choose **Add attribute** and enter the **Key** and **Value** details. This is where you will be mapping the attribute coming from your identity source to an attribute that IAM Identity Center passes as a session tag.

Key ⓘ	Value (optional) ⓘ	Remove
<input type="text" value="Department"/>	<input type="text" value="\${path:enterprise.department}"/>	✕
<input type="text" value="CostCenter"/>	<input type="text" value="\${path:enterprise.costCenter}"/>	✕
<input type="text" value="Add new key"/>	<input type="text" value="Add new value"/>	

Key represents the name you are giving to the attribute for use in policies. This can be any arbitrary name, but you need to specify that exact name in the policies you author for access control. For example, let's say that you are using Okta (an external IdP) as your identity source and need to pass your organization's cost center data along as session tags. In **Key**, you would enter a similarly matched name like **CostCenter** as your key name. It's important to note that whichever name you choose here, it must also be named exactly the same in your [aws:PrincipalTag condition key](#) (that is, "ec2:ResourceTag/CostCenter": "\${aws:PrincipalTag/CostCenter}").

Note

Use a single-value attribute for your key, for example, **Manager**. IAM Identity Center doesn't support multi-value attributes for ABAC, for example, **Manager**, **IT Systems**.

Value represents the content of the attribute coming from your configured identity source. Here you can enter any value from the appropriate identity source table listed in [Attribute mappings for AWS Managed Microsoft AD directory](#). For example, using the context provided in the above mentioned example, you would review the list of supported IdP attributes and determine that the closest match of a supported attribute would be **\${path:enterprise.costCenter}** and you would then enter it in the **Value** field. See the screenshot provided above for reference. Note, that you can't use external IdP attribute values outside of this list for ABAC unless you use the option of passing attributes through the SAML assertion.

5. Choose **Save changes**.

Now that you have configured mapping your access control attributes, you need to complete the ABAC configuration process. To do this, create your ABAC rules and add them to your permission sets and/or resource-based policies. This is required so that you can grant user identities access to AWS resources. For more information, see [Create permission policies for ABAC in IAM Identity Center](#).

Disable attributes for access control

Use the following procedure to disable the ABAC feature and delete all of the attribute mappings that have been configured.

To disable Attributes for access control

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**
3. On the **Settings** page, choose the **Attributes for access control** tab, and then choose **Disable**.
4. In the **Disable attributes for access control** dialog, review the information and when ready enter DELETE, and then choose **Confirm**.

Important

This step deletes all attributes that have been configured. Once deleted, any attributes that are received from an identity source and any custom attributes you have previously configured will not be passed.

Create permission policies for ABAC in IAM Identity Center

You can create permissions policies that determine who can access your AWS resources based on the configured attribute value. When you enable ABAC and specify attributes, IAM Identity Center passes the attribute value of the authenticated user into IAM for use in policy evaluation.

aws:PrincipalTag condition key

You can use access control attributes in your permission sets using the `aws:PrincipalTag` condition key for creating access control rules. For example, in the following trust policy you can

tag all the resources in your organization with their respective cost centers. You can also use a single permission set that grants developers access to their cost center resources. Now, whenever developers federate into the account using single sign-on and their cost center attribute, they only get access to the resources in their respective cost centers. As the team adds more developers and resources to their project, you only have to tag resources with the correct cost center. Then you pass cost center information in the AWS session when developers federate into AWS accounts. As a result, as the organization adds new resources and developers to the cost center, developers can manage resources aligned to their cost centers without needing any permission updates.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/CostCenter": "${aws:PrincipalTag/CostCenter}"
        }
      }
    }
  ]
}
```

For more information, see [aws:PrincipalTag](#) and [EC2: Start or stop instances based on matching principal and resource tags](#) in the *IAM User Guide*.

If policies contain invalid attributes in their conditions, then the policy condition will fail and access will be denied. For more information, see [Error 'An unexpected error has occurred' when a user tries to sign in using an external identity provider](#).

Repair the IAM identity provider

When you add single sign-on access to an AWS account, IAM Identity Center creates an IAM identity provider in each AWS account. An IAM identity provider helps keep your AWS account secure because you don't have to distribute or embed long-term security credentials, such as access keys, in your application.

If you delete or modify your identity provider, you must manually reapply your user and group assignments. Reapplying your user and group assignments recreates the identity provider. For more information, see:

- [Manage access to AWS accounts](#)
- [Manage access to applications](#)

Service-linked roles

[Service-linked roles](#) are predefined IAM permissions that allow IAM Identity Center to delegate and enforce which users have single sign-on access to specific AWS accounts in your organization in AWS Organizations. The service enables this functionality by provisioning a service-linked role in every AWS account within its organization. The service then allows other AWS services like IAM Identity Center to leverage those roles to perform service-related tasks. For more information, see [AWS Organizations and service-linked roles](#).

When you enable IAM Identity Center, IAM Identity Center creates a service-linked role in all accounts within the organization in AWS Organizations. IAM Identity Center also creates the same service-linked role in every account that is subsequently added to your organization. This role allows IAM Identity Center to access each account's resources on your behalf. For more information, see [Manage access to AWS accounts](#).

Service-linked roles that are created in each AWS account are named `AWSServiceRoleForSSO`. For more information, see [Using service-linked roles for IAM Identity Center](#).

Manage access to applications

With AWS IAM Identity Center, you can control who can have single sign-on access to your applications. Users get seamless access to these applications after they use their directory credentials to sign in.

IAM Identity Center securely communicates with these applications through a trusted relationship between IAM Identity Center and the application's service provider. This trust can be created in different ways, depending on the application type.

IAM Identity Center supports two application types: [AWS managed applications](#) and [customer managed applications](#). AWS managed applications are configured directly from within the relevant application consoles or through the application APIs. Customer managed applications must be added to the IAM Identity Center console and configured with the appropriate metadata for both IAM Identity Center and the service provider.

After you configure applications to work with IAM Identity Center, you can manage which users or groups access the applications. By default, no users are assigned to applications.

You can also grant your employees access to the AWS Management Console for a specific AWS account in your organization. For more information, see [Manage access to AWS accounts](#).

Topics

- [AWS managed applications](#)
- [Customer managed applications](#)
- [Trusted identity propagation across applications](#)
- [Manage IAM Identity Center certificates](#)
- [Application properties in the IAM Identity Center console](#)
- [Assign user access to applications in the IAM Identity Center console](#)
- [Remove user access in the IAM Identity Center console](#)
- [Map attributes in your application to IAM Identity Center attributes](#)

AWS managed applications










AWS managed applications integrate with IAM Identity Center and use it for authentication and directory services.

The integration of AWS managed applications with IAM Identity Center gives you an easier path to assign user access, without the need to set up separate federation or user and group synchronization for each application. You can [connect the identity source that you want to use](#) for authentication once, and you receive a single [view of user and group assignments](#). Administrators of the applications that enable trusted identity propagation are able to define and audit access to their application resources based on a user or the user's group membership, without the need to map them to IAM roles.






















AWS managed applications provide an administrative user interface that you can use to manage access to application resources. For example, QuickSight administrators can assign users to access dashboards based on their group membership. Most AWS managed applications also provide an AWS Management Console experience that enables you to assign users to the application. The console experience for these applications might integrate both functions, to combine user assignment capabilities with the ability to manage access to application resources.

AWS managed applications integrated with IAM Identity Center include:

AWS managed applications that integrate with IAM Identity Center

AWS managed application	Integrated with organization instance of IAM Identity Center	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center
Amazon Athena SQL	 Yes	 Yes	 Yes
Amazon CodeCatalyst	 Yes	 Yes	 No
Amazon EMR notebooks	 Yes	 No	 No

AWS managed application	Integrated with organization instance of IAM Identity Center	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center
Amazon EMR on Amazon EC2	 Yes	 Yes	 Yes
Amazon EMR Studio	 Yes	 Yes	 Yes
Amazon Kendra	 Yes	 No	 No
Amazon Managed Grafana	 Yes	 No	 No
Amazon Monitron	 Yes	 No	 No
Amazon Nimble Studio	 Yes	 No	 No
Amazon Pinpoint	 Yes	 No	 No
Amazon Q Business	 Yes	 Yes	 No

AWS managed application	Integrated with organization instance of IAM Identity Center	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center
Amazon Q Developer			
Amazon QuickSight			
Amazon Redshift			
Amazon S3 Access Grants			
Amazon SageMaker Studio			
Amazon WorkSpaces Web			
AWS CLI			

AWS managed application	Integrated with organization instance of IAM Identity Center	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center
AWS Deadline Cloud			
AWS IoT Events			
AWS IoT Fleet Hub			
AWS IoT SiteWise			
AWS Lake Formation			
AWS Supply Chain			
AWS Systems Manager			
AWS Verified Access			

* Account instances of IAM Identity Center are supported unless your users require access to the full set of Amazon Q Developer features on AWS websites. For more information, see [Setting up Amazon Q Developer](#) in the *Amazon Q Developer User Guide*.

Topics

- [Controlling access](#)
- [Coordinating administrative tasks](#)
- [Configuring IAM Identity Center to share identity information](#)
- [Considerations for sharing identity information in AWS accounts](#)
- [Enabling identity-aware console sessions](#)
- [Constraining the use of AWS managed applications](#)
- [Viewing and updating details about an AWS managed application](#)
- [Disabling an AWS managed application](#)

Controlling access

Access to AWS managed applications is controlled in two ways:

- **Initial entry to the application** – IAM Identity Center manages this through assignments to the application. By default, assignments are required for AWS managed applications.
- **Access to application resources** – The application manages this through independent resource assignments that it controls.

Coordinating administrative tasks

If you're an application administrator, you can choose whether to require assignments to an application. If assignments are required, when users sign in to the AWS access portal, only users who are assigned to the application directly or through a group assignment can view the application tile. Alternatively, if assignments aren't required, you can allow all IAM Identity Center users to enter the application. In this case, the application manages access to resources and the application tile is visible to all users who visit the AWS access portal.

If you're an IAM Identity Center administrator, you can use the IAM Identity Center console to remove assignments to AWS managed applications. Before you remove assignments, we recommend that you coordinate with the application administrator. You should also coordinate

with the application administrator if you plan to modify the setting that determines whether assignments required, or automate application assignments.

Configuring IAM Identity Center to share identity information

IAM Identity Center provides an identity store that contains user and group attributes, excluding sign-in credentials. You can use either of the following methods to keep the users and groups in your IAM Identity Center identity store updated:

- Use the IAM Identity Center identity store as your main identity source. If you choose this method, you manage your users, their sign-in credentials, and groups from within the IAM Identity Center console or AWS Command Line Interface (AWS CLI). For more information, see [Manage identities in IAM Identity Center](#).
- Set up provisioning (synchronization) of users and groups coming from either of the following identity sources to your IAM Identity Center identity store:
 - **Active Directory** – For more information, see [Connect to a Microsoft AD directory](#).
 - **External identity provider** – For more information, see [Manage an external identity provider](#).

If you choose this provisioning method, you continue managing your users and groups from within your identity source, and those changes are synchronized to the IAM Identity Center identity store.

Whichever identity source you choose, IAM Identity Center can share the user and group information with AWS managed applications. That way, you can connect an identity source to IAM Identity Center once and then share identity information with multiple applications in the AWS Cloud. This eliminates the need to independently set up federation and identity provisioning with each application. This sharing feature also makes it easy to give your users access to many applications in different AWS accounts.

Considerations for sharing identity information in AWS accounts

IAM Identity Center supports most commonly used attributes across applications. These attributes include first and last name, phone number, email address, address, and preferred language. Carefully consider which applications and which accounts can use this personally identifiable information.

You can control access to this information in either of the following ways. You can choose to enable access in only the AWS Organizations management account or in all accounts in AWS

Organizations. Or, you can use service control policies (SCPs) to control which applications can access the information in which accounts in AWS Organizations. For example, if you enable access in the AWS Organizations management account only, then applications in member accounts have no access to the information. However, if you enable access in all accounts, you can use SCPs to disallow access by all applications except those you want to permit.

Enabling identity-aware console sessions

An identity-aware session for the console enhances a user's AWS console session by providing some additional user context to personalize that user's experience. This capability is currently supported for Amazon Q Developer Pro users of [Amazon Q on AWS apps and websites](#).

You can enable identity-aware console sessions without making any changes to existing access patterns or federation into the AWS console. If your users sign in to the AWS console with IAM (for example, if they sign in as IAM users or through federated access with IAM), they can continue using these methods. If your users sign in to the AWS access portal, they can continue using their IAM Identity Center user credentials.

Topics

- [Prerequisites and considerations](#)
- [How to enable identity-aware-console sessions](#)
- [How identity-aware console sessions work](#)

Prerequisites and considerations

Before you enable identity-aware console sessions, review the following prerequisites and considerations:

- If your users access Amazon Q on AWS apps and websites through an Amazon Q Developer Pro subscription, you must enable identity-aware console sessions.

Note

Amazon Q Developer users can access Amazon Q without identity-aware sessions, but they won't have access to their Amazon Q Developer Pro subscriptions.

- Identity-aware console sessions require an [organization instance](#) of IAM Identity Center.

- Integration with Amazon Q isn't supported if you enable IAM Identity Center in an opt-in AWS Region.
- After you enable identity-aware console sessions, you can't disable this capability.
- To enable identity-aware console sessions, you must have the following permissions:
 - `sso:CreateApplication`
 - `sso:GetSharedSsoConfiguration`
 - `sso:ListApplications`
 - `sso:PutApplicationAssignmentConfiguration`
 - `sso:PutApplicationAuthenticationMethod`
 - `sso:PutApplicationGrant`
 - `sso:PutApplicationAccessScope`
 - `signin:CreateTrustedIdentityPropagationApplicationForConsole`
 - `signin:ListTrustedIdentityPropagationApplicationsForConsole`
 -
- To enable your users to use identity-aware console sessions, you must grant them the `sts:setContext` permission in an identity-based policy. For information, see [Granting users permissions to use identity-aware console sessions](#).

How to enable identity-aware-console sessions

You can enable identity-aware console sessions in the Amazon Q console or in the IAM Identity Center console.

Enable identity-aware console sessions in the Amazon Q console

Before you enable identity-aware console sessions, you must have an organization instance of IAM Identity Center with an identity source connected. If you've already configured IAM Identity Center, skip to step 3.

1. Open the IAM Identity Center console. Choose **Enable**, and create an organization instance of IAM Identity Center. For information, see [Enabling AWS IAM Identity Center](#).
2. Connect your identity source to IAM Identity Center and provision users into IAM Identity Center. You can connect your existing identity source to IAM Identity Center or use the Identity Center directory if you're not already using another identity source. For more information, see [Getting started tutorials for IAM Identity Center](#).

3. After you finish setting up IAM Identity Center, open the Amazon Q console and follow the steps in [Subscriptions](#) in the *Amazon Q Developer User Guide*. Make sure to enable identity-aware console sessions.

Note

If you don't have sufficient permissions to enable identity-aware console sessions, you might need to ask an IAM Identity Center administrator to perform this task for you in the IAM Identity Center console. For more information, see the next procedure.

Enable identity-aware console sessions in the IAM Identity Center console

If you're an IAM Identity Center administrator, you might be asked by another administrator to enable identity-aware console sessions in the IAM Identity Center console.

1. Open the IAM Identity Center console.
2. In the navigation pane, choose **Settings**.
3. Under **Enable identity-aware sessions**, choose **Enable**.
4. In the second message, choose **Enable**.
5. After you finish enabling identity-aware console sessions, a confirmation message appears at the top of the **Settings** page.
6. In the **Details** section, the status for **Identity-aware sessions** is **Enabled**.

How identity-aware console sessions work

IAM Identity Center enhances a user's current console session to include the active IAM Identity Center user's ID and the IAM Identity Center session ID.

Identity-aware console sessions include the following three values:

- **Identity store user ID** ([identitystore:UserId](#)) - This value is used to uniquely identify a user in the identity source that is connected to IAM Identity Center.
- **Identity store directory ARN** ([identitystore:IdentityStoreArn](#)) - This value is the ARN of the identity store that is connected to IAM Identity Center, and where you can look up attributes for `identitystore:UserId`.

- **IAM Identity Center session ID** - This value indicates whether the user's IAM Identity Center session is still valid.

The values are the same, but obtained in different ways and added at different points of the process, depending on how the user signs in:

- **IAM Identity Center (AWS access portal):** In this case, the user's identity store user ID and ARN values are already provided in the active IAM Identity Center session. IAM Identity Center enhances the current session by adding only the session ID.
- **Other sign-in methods:** If the user signs in to AWS as an IAM user, with an IAM role, or as a federated user with IAM, none of these values are provided. IAM Identity Center enhances the current session by adding the identity store user ID, identity store directory ARN, and the session ID.

Constraining the use of AWS managed applications

When you enable IAM Identity Center for the first time, AWS allows the use of AWS managed applications automatically in all accounts in AWS Organizations. To constrain applications, you must implement SCPs. You can use SCPs to block access to the IAM Identity Center user and group information and to prevent the application from being started, except in designated accounts.

Viewing and updating details about an AWS managed application

After you connect an AWS managed application to IAM Identity Center by using the console or APIs for the application, the application is registered with IAM Identity Center. After an application is registered with IAM Identity Center, you can view and update detailed information about the application in the IAM Identity Center console.

To view information about an AWS managed application in the IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **AWS managed** tab.
4. Choose the link for the managed application you'd like to open and view.
5. Information about the application includes whether user and group assignments are required, and if applicable, assigned users and groups and trusted applications for identity propagation.

For information about trusted identity propagation, see [Trusted identity propagation across applications](#).

To update information about an AWS managed application in the IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **AWS managed** tab.
4. Choose the link for the managed application you'd like to open and update.
5. Choose **Action** and then choose **Edit Details**.
6. You can change the application's display name, description, as well as the user and group assignment method.
 - a. To change the display name, enter the desired name in the **Display name** field and choose **Save changes**.
 - b. To change the description, enter the desired description in the **Description** field and choose **Save changes**.
 - c. To change the user and group assignment method, make the desired change and choose **Save changes**. For more information, see [the section called "Users, groups, and provisioning"](#).

Disabling an AWS managed application

To prevent users from authenticating to an AWS managed application, you can disable the application in the IAM Identity Center console.

Warning

Disabling an application deletes all user permissions to this application, disconnects the application from IAM Identity Center, and renders the application inaccessible. If you're an IAM Identity Center administrator, we recommend that you coordinate with the application administrator before performing this task.

To disable an AWS managed application

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. On the **Applications** page, under **AWS managed applications**, choose the application that you want to disable.
4. With the application selected, choose **Actions**, and then choose **Disable**.
5. In the **Disable application** dialog box, choose **Disable**.
6. In the **AWS managed applications** list, the application status appears as **Inactive**.

Customer managed applications

With IAM Identity Center, you can create or connect workforce users and centrally manage their access across all their AWS accounts and applications. IAM Identity Center acts as a central identity service and provides different ways for your users to be authenticated. If you already use an identity provider (IdP), IAM Identity Center can integrate with your IdP so that you can provision your users and groups into IAM Identity Center and use your IdP for authentication.

If you use customer managed applications that support [SAML 2.0](#), you can federate your IdP to IAM Identity Center through SAML 2.0 and use IAM Identity Center to manage user access to those applications. IAM Identity Center provides a catalog of commonly used applications that support SAML 2.0, such as Salesforce and Microsoft 365. This catalog is available in the IAM Identity Center console. You can also set up your own SAML 2.0 applications.

Note

If you have customer managed applications that support OAuth 2.0 and your users need access from these applications to AWS services, you can use trusted identity propagation. With trusted identity propagation, a user can sign in to an application, and that application can pass the users' identity in requests to access data in AWS services. For more information, see [Using trusted identity propagation with customer managed applications](#).

Topics

- [Single sign-on access to SAML 2.0 and OAuth 2.0 applications](#)

- [Setting up customer managed SAML 2.0 applications](#)

Single sign-on access to SAML 2.0 and OAuth 2.0 applications

IAM Identity Center enables you to provide your users with single sign-on access to SAML 2.0 or OAuth 2.0 applications. The following topics provide a high-level overview of SAML 2.0 and OAuth 2.0.

Topics

- [SAML 2.0](#)
- [OAuth 2.0](#)

SAML 2.0

SAML 2.0 is an industry standard used for securely exchanging SAML assertions that pass information about a user between a SAML authority (called an identity provider or IdP), and a SAML 2.0 consumer (called a service provider or SP). IAM Identity Center uses this information to provide federated single sign-on access for those users who are authorized to use applications within the AWS access portal.

OAuth 2.0

OAuth 2.0 is a protocol that allows applications to access and share user data securely without sharing passwords. This capability provides a secure and standardized way for users to allow applications access to their resources. Access is facilitated by different OAuth 2.0 grant flows.

IAM Identity Center enables applications that run on public clients to retrieve temporary credentials to access AWS accounts and services programmatically on behalf of their users. Public clients are typically desktops, laptops, or other mobile devices that are used to run applications locally. Examples of AWS applications that run on public clients include the AWS Command Line Interface (AWS CLI), AWS Toolkit, and AWS Software Development Kits (SDKs). To enable these applications to obtain credentials, IAM Identity Center supports portions of the following OAuth 2.0 flows:

- Authorization Code Grant with Proof Key for Code Exchange (PKCE) ([RFC 6749](#) and [RFC 7636](#))
- Device Authorization Grant ([RFC 8628](#))

Note

These grant types can be used only with AWS services that support this capability. These services may not support this grant type in all AWS Regions. Refer to the documentation of relevant AWS services for regional differences.

OpenID Connect (OIDC) is an authentication protocol that is based on the OAuth 2.0 Framework. OIDC specifies how to use OAuth 2.0 for authentication. Through the [IAM Identity Center OIDC service APIs](#), an application registers an OAuth 2.0 client and uses one of these flows to obtain an access token that provides permissions to IAM Identity Center protected APIs. An application specifies [access scopes](#) to declare its intended API user. After you, as the IAM Identity Center administrator, configure your identity source, your application end users must complete a sign-in process, if they have not already done so. Your end users must then provide their consent to allow the application to make API calls. These API calls are made using the users' permissions. In response, IAM Identity Center returns an access token to the application that contains the access scopes to which the users consented.

Using an OAuth 2.0 grant flow

OAuth 2.0 grant flows are only available through AWS managed applications that support the flows. To use an OAuth 2.0 flow, your instance of IAM Identity Center and any supported AWS managed applications that you use must be deployed in a single AWS Region. Refer to the documentation for each AWS service to determine the regional availability of AWS managed applications and the instance of IAM Identity Center that you want to use.

To use an application that uses an OAuth 2.0 flow, the end user must enter the URL where the application will connect and register with your instance of IAM Identity Center. Depending on the application, as the administrator, you must provide your users with the **AWS access portal URL** or the **Issuer URL** of your instance of IAM Identity Center. You can find these two settings on the [IAM Identity Center console Settings](#) page. For additional information about configuring a client application, refer to that application's documentation.

The end user experience for signing into an application and providing consent depends on whether the application uses the [Authorization Code Grant with PKCE](#) or [Device Authorization Grant](#).

Authorization Code Grant with PKCE

This flow is used by applications that run on a device that has a browser.

1. A browser window opens.
2. If the user has not authenticated, the browser redirects them to complete user authentication.
3. After authentication, the user is presented with a consent screen that displays the following information:
 - The name of the application
 - The access scopes that the application is requesting consent to use
4. The user can cancel the consent process or they can give their consent and the application proceeds with access based on the user's permissions.

Device Authorization Grant

This flow can be used by applications that run on a device with or without a browser. When the application initiates the flow, the application presents a URL and a user code that the user must verify later in the flow. The user code is necessary because the application that initiates the flow might be running on a different device than the device on which the user provides consent. The code ensures that the user is consenting to the flow they initiated on the other device.

1. When the flow initiates from a device with a browser, a browser window opens. When the flow initiates from a device without a browser, the user must open a browser on a different device and go to the URL that the application presented.
2. In either case, if the user has not authenticated, the browser redirects them to complete user authentication.
3. After authentication, the user is presented with a consent screen that displays the following information:
 - The name of the application
 - The access scopes that the application is requesting consent to use
 - The user code that the application presented to the user
4. The user can cancel the consent process or they can give their consent and the application proceeds with access based on the user's permissions.

Access scopes

A *scope* defines the access for a service for a service that can be accessed through an OAuth 2.0 flow. Scopes are a way for the service, also called a resource server, to group permissions related to actions and the service resources, and they specify the coarse-grained operations that OAuth 2.0

clients can request. When an OAuth 2.0 client registers with the [IAM Identity Center OIDC service](#), the client specifies the scopes to declare its intended actions, for which the user must provide consent.

OAuth 2.0 clients use scope values as defined in [section 3.3 of OAuth 2.0 \(RFC 6749\)](#) to specify what permissions are being requested for an access token. Clients can specify a maximum of 25 scopes when requesting an access token. When a user provides consent during an Authorization Code Grant with PKCE or Device Authorization Grant flow, IAM Identity Center encodes the scopes into the access token it returns.

AWS adds scopes to IAM Identity Center for supported AWS services. The following table lists the scopes that the IAM Identity Center OIDC service supports when you register a public client.

Access scopes supported by the IAM Identity Center OIDC service when registering a public client

Scope	Description	Services supported by
<code>sso:account:access</code>	Access IAM Identity Center managed accounts and permission sets.	IAM Identity Center
<code>codewhisperer:analysis</code>	Enable access to Amazon Q Developer code analysis.	AWS Builder ID and IAM Identity Center
<code>codewhisperer:completions</code>	Enable access to Amazon Q inline code suggestions.	AWS Builder ID and IAM Identity Center
<code>codewhisperer:conversations</code>	Enable access to Amazon Q chat.	AWS Builder ID and IAM Identity Center
<code>codewhisperer:taskassist</code>	Enable access to Amazon Q Developer Agent for software development.	AWS Builder ID and IAM Identity Center

Scope	Description	Services supported by
<code>codewhisperer:transformations</code>	Enable access to Amazon Q Developer Agent for code transformation.	AWS Builder ID and IAM Identity Center
<code>codecatalyst:read_write</code>	Read and write to your Amazon CodeCatalyst resources, allowing access to all your existing resources.	AWS Builder ID and IAM Identity Center

Setting up customer managed SAML 2.0 applications

If you use customer managed applications that support [SAML 2.0](#), you can federate your IdP to IAM Identity Center through SAML 2.0 and use IAM Identity Center to manage user access to those applications. You can select a SAML 2.0 application from a catalog of commonly used applications in the IAM Identity Center console, or you can set up your own SAML 2.0 application.

Note

If you have customer managed applications that support OAuth 2.0 and your users need access from these applications to AWS services, you can use trusted identity propagation. With trusted identity propagation, a user can sign in to an application, and that application can pass the users' identity in requests to access data in AWS services. For more information, see [Using trusted identity propagation with customer managed applications](#).

Topics

- [Set up an application from the IAM Identity Center application catalog](#)
- [Set up your own SAML 2.0 application](#)

Set up an application from the IAM Identity Center application catalog

You can use the application catalog in the IAM Identity Center console to add many commonly used SAML 2.0 applications that work with IAM Identity Center. Examples include Salesforce, Box, and Microsoft 365.

Most applications provide detailed information about how to set up the trust between IAM Identity Center and the application's service provider. This information is available in the configuration page for the application, after you select the application in the catalog. After you configure the application, you can assign access to users or groups in IAM Identity Center as needed.


Use this procedure to set up a SAML 2.0 trust relationship between IAM Identity Center and your application's service provider.

Before you begin this procedure, it's helpful to have the service provider's metadata exchange file so that you can more efficiently set up the trust. If you don't have this file, you can still use this procedure to configure the trust it manually.

To add and configure an application from the application catalog

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **Customer managed** tab.
4. Choose **Add application**.
5. On the **Select application type** page, under **Setup preference**, choose **I want to select an application from the catalog**.
6. Under **Application catalog**, start typing the name of the application that you want to add in the search box.
7. Choose the name of the application from the list when it appears in the search results, and then choose **Next**.
8. On the **Configure application** page, the **Display name** and **Description** fields are prepopulated with relevant details for the application. You can edit this information.
9. Under **IAM Identity Center metadata**, do the following:
 - a. Under **IAM Identity Center SAML metadata file**, choose **Download** to download the identity provider metadata.

- b. Under **IAM Identity Center certificate**, choose **Download certificate** to download the identity provider certificate.

 **Note**

You will need these files later when you set up the application from the service provider's website. Follow the instructions from that provider.

10. (Optional) Under **Application properties**, you can specify the **Application start URL**, **Relay state**, and **Session duration**. For more information, see [Application properties in the IAM Identity Center console](#).
11. Under **Application metadata**, do one of the following:
 - a. If you have a metadata file, choose **Upload application SAML metadata file**. Then, select **Choose file** to find and select the metadata file.
 - b. If you don't have a metadata file, choose **Manually type your metadata values**, and then provide the **Application ACS URL** and **Application SAML audience** values.
12. Choose **Submit**. You're taken to the details page of the application that you just added.

Set up your own SAML 2.0 application

You can set up your own applications that allow identity federation using SAML 2.0 and add them to IAM Identity Center. Most of the steps for setting up your own SAML 2.0 applications are the same as setting up a SAML 2.0 application from the application catalog in the IAM Identity Center console. However, you must also provide additional SAML attribute mappings for your own SAML 2.0 applications. These mappings enable IAM Identity Center to populate the SAML 2.0 assertion correctly for your application. You can provide this additional SAML attribute mapping when you set up the application for the first time. You can also provide SAML 2.0 attribute mappings on the application details page in the IAM Identity Center console.

Use the following procedure to set up a SAML 2.0 trust relationship between IAM Identity Center and your SAML 2.0 application's service provider. Before you begin this procedure, make sure that you have the service provider's certificate and metadata exchange files so that you can finish setting up the trust.

To set up your own SAML 2.0 application

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **Customer managed** tab.
4. Choose **Add application**.
5. On the **Select application type** page, under **Setup preference**, choose **I have an application I want to set up**.
6. Under **Application type**, choose **SAML 2.0**.
7. Choose **Next**.
8. On the **Configure application** page, under **Configure application**, enter a **Display name** for the application, such as **MyApp**. Then, enter a **Description**.
9. Under **IAM Identity Center metadata**, do the following:
 - a. Under **IAM Identity Center SAML metadata file**, choose **Download** to download the identity provider metadata.
 - b. Under **IAM Identity Center certificate**, choose **Download** to download the identity provider certificate.

Note

You will need these files later when you set up the custom application from the service provider's website.

10. (Optional) Under **Application properties**, you can also specify the **Application start URL**, **Relay state**, and **Session duration**. For more information, see [Application properties in the IAM Identity Center console](#).
11. Under **Application metadata**, choose **Manually type your metadata values**. Then, provide the **Application ACS URL** and **Application SAML audience** values.
12. Choose **Submit**. You're taken to the details page of the application that you just added.

Trusted identity propagation across applications

Trusted identity propagation enables AWS services to do the following:

- Authorize access to AWS resources based on the user's identity context.
- Securely share the user's identity context with other AWS services.

These capabilities enable user access to be more easily defined, granted, and logged.

With trusted identity propagation, a user can sign in to an application, and that application can pass the users' identity context in requests to access data in AWS services. Because access is managed based on a user's identity, users don't need to use database local user credentials or assume an IAM role to access data.

Topics

- [Trusted identity propagation overview](#)
- [Trusted identity propagation use cases](#)
- [Set up trusted identity propagation](#)
- [Using applications with a trusted token issuer](#)

Trusted identity propagation overview

With trusted identity propagation, user access to AWS resources can be more easily defined, granted, and logged. Trusted identity propagation is built on the [OAuth 2.0 Authorization Framework](#), which allows applications to access and share user data securely without sharing passwords. OAuth 2.0 provides secure delegated access to application resources. Access is *delegated* because the resource administrator approves, or delegates the application that the user signs in to, to access the other application.

To avoid sharing user passwords, trusted identity propagation uses tokens. Tokens provide a standard way for a trusted application to claim who the user is and what requests are permitted between two applications. AWS managed applications that integrate with trusted identity propagation obtain tokens from IAM Identity Center directly. IAM Identity Center also provides an option for applications to exchange identity tokens and access tokens that come from an external OAuth 2.0 authorization server. This makes it possible for an application to authenticate and obtain tokens outside of AWS, exchange the token for an IAM Identity Center token, and use the new token to make requests to AWS services. For more information, see [Using applications with a trusted token issuer](#).

The OAuth 2.0 process starts when a user signs in to an application. The application that the user signs in to initiates a request to access the other application's resources. The initiating (requesting)

application can access the receiving application on behalf of the user by requesting a token from the authorization server. The authorization server returns the token, and the initiating application passes that token, with a request for access, to the receiving application.

Trusted identity propagation use cases

As an IAM Identity Center administrator, you might be asked to help configure trusted identity propagation between the following initiating applications that support this capability and connected AWS services. The following sections provide more information about the specific use cases supported by applications that can initiate trusted identity propagation.

Topics

- [Amazon EMR](#)
- [Amazon QuickSight](#)
- [Amazon Redshift query editor v2](#)
- [Third-party business intelligence applications](#)
- [Custom-developed applications](#)

Amazon EMR

You can use Amazon EMR as the initiating application for the following trusted identity propagation use cases.

Description	Other AWS services used	Learn more
Run interactive analyses with Apache Spark on Amazon EMR on Amazon EC2 clusters through Amazon EMR Studio. Apply access control based on workforce identities and associated attributes for AWS Glue Catalog through AWS Lake Formation.	Amazon EMR on Amazon EC2 authorized through AWS Lake Formation, Amazon S3 Access Grants, Amazon S3, AWS Service Catalog	<ul style="list-style-type: none"> • Integrate Amazon EMR with IAM Identity Center in the <i>Amazon EMR Management Guide</i>. • Amazon S3 Access Grants and corporate directory identities in the <i>Amazon Simple Storage Service User Guide</i>. • Connecting AWS Lake Formation with IAM Identity

Description	Other AWS services used	Learn more
	<p data-bbox="662 289 792 331">Note</p> <ul data-bbox="711 373 954 1054" style="list-style-type: none"><li data-bbox="711 373 954 592">• Requires access through Amazon EMR Studio.<li data-bbox="711 625 954 751">• Table-level access control only.<li data-bbox="711 785 954 1054">• Apache Hive, PrestoSQL /Trino, and EMR Serverless are not supported.	<p data-bbox="1052 256 1448 340">Center in the <i>AWS Lake Formation Developer Guide</i></p> <ul data-bbox="1026 361 1513 541" style="list-style-type: none"><li data-bbox="1026 361 1513 541">• Use your corporate identities for analytics with Amazon EMR and IAM Identity Center in the <i>AWS Big Data Blog</i>

Description	Other AWS services used	Learn more
<p>Run adhoc analyses with Trino on Athena through Amazon EMR Studio. Apply access control based on workforce identities and associated attributes for AWS Glue Catalog through AWS Lake Formation. Secure access to an Athena query result bucket location in Amazon S3 by using Amazon S3 Access Grants.</p>	<p>Athena authorized through AWS Lake Formation, Amazon S3 Access Grants</p> <div data-bbox="634 495 987 999" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>Requires access through Amazon EMR Studio. Direct access from the Amazon Athena console is not supported.</p> </div>	<ul style="list-style-type: none"> • Integrate Amazon EMR with IAM Identity Center in the <i>Amazon EMR Management Guide</i>. • Using IAM Identity Center enabled Athena workgroups in the <i>Amazon Athena User Guide</i>. • Amazon S3 Access Grants and corporate directory identities in the <i>Amazon Simple Storage Service User Guide</i>. • Connecting AWS Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>. • Bring your workforce identity to Amazon EMR Studio and Athena in the <i>AWS Big Data Blog</i>.

Amazon QuickSight

You can use Amazon QuickSight as the initiating application for the following trusted identity propagation use cases.

Description	Other AWS services used	Learn more
<p>Amazon QuickSight users can query Amazon Redshift data. Data access is granted in Amazon</p>	<p>Amazon Redshift</p>	<ul style="list-style-type: none"> • Connect Redshift with IAM Identity Center to give users a single sign-on experience

Description	Other AWS services used	Learn more
Redshift by an Amazon Redshift administrator.		<p>e in the <i>Amazon Redshift Management Guide</i>.</p> <ul style="list-style-type: none"> • Connect Amazon Redshift with IAM Identity Center through Amazon QuickSight in the <i>Amazon Redshift Management Guide</i>.
Amazon QuickSight users can query Amazon Redshift Spectrum for structured data in Amazon S3, with access that is authorized by an AWS Lake Formation administrator.	<p>Amazon Redshift Spectrum, Amazon S3 structured data</p> <p>*Through Amazon Redshift Spectrum authorized through AWS Lake Formation</p>	<ul style="list-style-type: none"> • Connect Redshift with IAM Identity Center to give users a single sign-on experience in the <i>Amazon Redshift Management Guide</i>. • Connect Amazon Redshift with IAM Identity Center through Amazon QuickSight in the <i>Amazon Redshift Management Guide</i>. • Connecting AWS Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>. • Simplify access management with Amazon Redshift and AWS Lake Formation for users in an External Identity Provider in the <i>AWS Big Data Blog</i>.

Description	Other AWS services used	Learn more
<p>Amazon QuickSight users can query Amazon Redshift datashares for structured data in Amazon S3, with access that is authorized by an AWS Lake Formation administrator.</p>	<p>Amazon Redshift datashares, Amazon S3 structured data</p> <p>*Through Amazon Redshift authorized through AWS Lake Formation</p>	<ul style="list-style-type: none"> • Connect Amazon Redshift with IAM Identity Center through Amazon QuickSight in the <i>Amazon Redshift Management Guide</i>. • Connecting AWS Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>. • Simplify access management with Amazon Redshift and AWS Lake Formation for users in an External Identity Provider in the <i>AWS Big Data Blog</i>.

Amazon Redshift query editor v2

You can use Amazon Redshift query editor v2 as the initiating application for the following trusted identity propagation use cases.

Description	Other AWS services used	Learn more
<p>Amazon Redshift query editor v2 users can query Amazon Redshift data. Data access is granted in Amazon Redshift by an Amazon Redshift administrator.</p>	<p>Amazon Redshift</p>	<ul style="list-style-type: none"> • Connect Redshift with IAM Identity Center to give users a single sign-on experience in the <i>Amazon Redshift Management Guide</i>. • Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i>.

Description	Other AWS services used	Learn more
		<ul style="list-style-type: none"> • Integrate Okta with Amazon Redshift Query Editor V2 using AWS IAM Identity Center for seamless Single Sign-on in the <i>AWS Big Data Blog</i>.
<p>Amazon Redshift query editor v2 users can query Amazon Redshift Spectrum external tables for structured data in Amazon S3, with access that is authorized by an AWS Lake Formation administrator.</p>	<p>Amazon Redshift Spectrum, Amazon S3 structured data</p> <p>*Through Amazon Redshift Spectrum authorized through AWS Lake Formation</p>	<ul style="list-style-type: none"> • Connect Redshift with IAM Identity Center to give users a single sign-on experience in the <i>Amazon Redshift Management Guide</i>. • Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i>. • Connecting AWS Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>.
<p>Amazon Redshift query editor v2 users can query Amazon Redshift datashares with access that is authorized by an AWS Lake Formation administrator.</p>	<p>Amazon Redshift datashares, AWS Lake Formation</p>	<ul style="list-style-type: none"> • Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i>. • Connecting AWS Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>.

Third-party business intelligence applications

You can use a third-party business intelligence application such as Tableau, as the initiating application for specific trusted identity propagation use cases. Modified third-party business

intelligence applications can pass the Amazon Redshift driver the identity of a user through OAuth identity tokens or access tokens, to query Amazon Redshift for data, with access that is authorized by an Amazon Redshift administrator.

Tableau

You can use Tableau Desktop, Tableau Server, and Tableau Prep as the initiating applications for the following trusted identity propagation use cases.

Description	Other AWS services used	Learn more
<p>Tableau users can query Amazon Redshift data. Data access is granted in Amazon Redshift by an Amazon Redshift administrator.</p>	<p>Amazon Redshift</p>	<ul style="list-style-type: none"> • Integrate Tableau and Okta with Amazon Redshift using IAM Identity Center in the <i>AWS Big Data blog</i> • Enabling IAM Identity Center in <i>IAM Identity Center User Guide</i> • Connect IAM Identity Center with your preferred IdP and sync users and groups <i>IAM Identity Center User Guide</i> • Connect Amazon Redshift with IAM Identity Center to give users a single sign-on experience in <i>Amazon Redshift Management Guide</i> • Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i> • Integrate Identity Provider (IdP) with Amazon Redshift Query Editor V2 using AWS IAM Identity Center for

Description	Other AWS services used	Learn more
		<p>seamless Single Sign-On in the <i>AWS Big Data blog</i></p> <ul style="list-style-type: none">• the section called “Trusted token issuer” in <i>IAM Identity Center User Guide</i>

Description	Other AWS services used	Learn more
<p>Tableau users can query Amazon Redshift Spectrum external tables for structured data in Amazon S3, with access control based on workforce identities and associated attributes for AWS Glue Data Catalog through AWS Lake Formation.</p>	<p>Amazon Redshift Spectrum, Amazon S3 structured data</p> <p>*Through Amazon Redshift Spectrum authorized via AWS Lake Formation</p>	<ul style="list-style-type: none"> • Integrate Tableau and Okta with Amazon Redshift using IAM Identity Center in the <i>AWS Big Data blog</i> • Enabling IAM Identity Center in the <i>IAM Identity Center User Guide</i> • Connect IAM Identity Center with your preferred IdP and sync users and groups in the <i>IAM Identity Center User Guide</i> • Connect Redshift with IAM Identity Center to give users a single sign-on experience in the <i>Amazon Redshift Management Guide</i> • Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i> • Integrate Identity Provider (IdP) with Amazon Redshift Query Editor V2 using AWS IAM Identity Center for seamless Single Sign-On in the <i>AWS Big Data blog</i> • Connecting Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>

Description	Other AWS services used	Learn more
		<ul style="list-style-type: none">• the section called “Trusted token issuer” in the <i>IAM Identity Center User Guide</i>

Description	Other AWS services used	Learn more
<p>Tableau users can query Amazon Redshift datashares with access control based on workforce identities and associated attributes for AWS Glue Data Catalog through AWS Lake Formation.</p>	<p>Amazon Redshift datashares, AWS Lake Formation</p>	<ul style="list-style-type: none">• Integrate Tableau and Okta with Amazon Redshift using IAM Identity Center in the <i>AWS Big Data blog</i>• Enabling IAM Identity Center in the <i>IAM Identity Center User Guide</i>• Connect IAM Identity Center with your preferred IdP and sync users and groups in the <i>IAM Identity Center User Guide</i>• Connect to an Amazon Redshift database in the <i>Amazon Redshift Management Guide</i>• Integrate Identity Provider (IdP) with Amazon Redshift Query Editor V2 using AWS IAM Identity Center for seamless Single Sign-On in the <i>AWS Big Data blog</i>• Connecting Lake Formation with IAM Identity Center in the <i>AWS Lake Formation Developer Guide</i>• Simplify access management with Amazon Redshift and AWS Lake Formation for users in an External Identity Provider in the <i>AWS Big Data Blog</i>

Description	Other AWS services used	Learn more
		<ul style="list-style-type: none"> • the section called “Trusted token issuer” in the <i>IAM Identity Center User Guide</i>

Custom-developed applications

You can use your own custom-developed applications as an initiating application for the following trusted identity propagation use cases.

Description	Other AWS services used	Learn more
<p>Create an application that authenticates users through an OAuth authorization server, then use AWS IAM Identity Center and IAM to obtain an identity-enhanced IAM role credential. This credential is used to request access to unstructured data in Amazon S3, with access that is authorized by an Amazon S3 Access Grants administrator.</p>	<p>AWS IAM Identity Center, Amazon S3 unstructured data</p> <p>*Authorized through Amazon S3 Access Grants</p>	<ul style="list-style-type: none"> • Amazon S3 Access Grants and corporate directory identities in the <i>Amazon Simple Storage Service User Guide</i>. • How to develop a user-facing data application with IAM Identity Center and Amazon S3 Access Grants (Part 1) and (Part 2) in the <i>AWS Storage Blog</i>.
<p>Build a custom application that interacts with Amazon Q Business to respond to user questions based on your own content and the user's permissions.</p>	<p>IAM Identity Center, Amazon Q Business</p>	<ul style="list-style-type: none"> • Enable and configure an IAM Identity Center instance in the <i>Amazon Q Business User Guide</i>. • How to use AWS managed applications with IAM Identity Center: Enable Amazon Q without migrating existing IAM federation flows in the <i>AWS Security Blog</i>.

Description	Other AWS services used	Learn more
		<ul style="list-style-type: none">• Build a custom UI for Amazon Q in the <i>AWS Machine Learning Blog</i>.

Set up trusted identity propagation

Trusted identity propagation supports different ways for applications to authenticate so that they can pass a user's identity to AWS services. Setup for trusted identity propagation varies, based on the application types and how they authenticate.

Note

You must [set up a trusted token issuer](#) if you have customer managed applications that request access to AWS managed applications, but don't use AWS APIs to connect.

Topics

- [Prerequisites and considerations](#)
- [Using trusted identity propagation with AWS managed applications](#)
- [Using trusted identity propagation with customer managed applications](#)

Prerequisites and considerations

Before you set up trusted identity propagation, review the following prerequisites and considerations.

Topics

- [Prerequisites](#)
- [Additional considerations](#)

Prerequisites

To use trusted identity propagation, ensure that your environment meets the following prerequisites.

- **IAM Identity Center deployment with users and groups provisioned**

To use trusted identity propagation, you must enable IAM Identity Center and provision users and groups. For information, see [Get started with common tasks in IAM Identity Center](#).

Organization instance recommended – We recommend that you use an [organization instance](#) of IAM Identity Center that you enable in the management account of AWS Organizations. If you plan to use trusted identity propagation to enable users to access AWS services and related resources in different AWS accounts within the same organization, you can [delegate administration](#) of your instance of IAM Identity Center to a member account.

If you plan to use a single [account instance](#) of IAM Identity Center, all AWS services and resources that you want users to access through trusted identity propagation must reside in the same standalone AWS account, or in the same member account in the organization where you enabled IAM Identity Center. For more information, see [Account instances of IAM Identity Center](#).

- **For AWS managed applications; connection to IAM Identity Center**

To use trusted identity propagation, AWS managed applications must integrate with IAM Identity Center.

Additional considerations

Keep in mind the following additional considerations for using trusted identity propagation.

- **Don't modify the Require assignments setting for AWS managed applications**

AWS managed applications have a default setting configuration that determines whether assignments are required for users and groups. We recommend that you do not modify this setting. Even if you have configured fine-grained permissions that allow user access to specific resources, modifying the **Require assignments** setting might result in unexpected behavior, including disrupted user access to these resources.

- **Multi-account permissions (permission sets) not required**

Trusted identity propagation doesn't require you to set up [multi-account permissions](#) (permission sets). You can enable IAM Identity Center and use it for trusted identity propagation only.

Using trusted identity propagation with AWS managed applications

Trusted identity propagation enables an AWS managed application to request access to data in AWS services on behalf of a user. Data access management is based on a user's identity, so administrators can grant access based on users' existing user and group memberships. The user's identity, actions performed on their behalf, and other events are recorded in service-specific logs and CloudTrail events.

Trusted identity propagation is based on the OAuth 2.0 standard. To use this capability, AWS managed applications must integrate with IAM Identity Center. AWS analytics services might provide driver-based interfaces that enable a compatible application to use trusted identity propagation. For example, JDBC, ODBC, and Python drivers enable compatible query tools to use trusted identity propagation without the need for you to complete additional setup steps.

Topics

- [Set up AWS managed applications for trusted identity propagation](#)
- [Trusted identity propagation request flows for AWS managed applications](#)
- [After an application obtains a token](#)
- [Identity-enhanced IAM role sessions](#)
- [Types of identity-enhanced IAM role sessions](#)
- [Identity-enhanced IAM role session logging](#)
- [Setup process and request flow for AWS managed applications](#)

Set up AWS managed applications for trusted identity propagation

AWS services that support trusted identity propagation provide an administrative user interface and APIs that you can use to set up this capability. No configuration is required within IAM Identity Center for these services.

Following is the high-level process for setting up an AWS service for trusted identity propagation. The specific steps vary depending on the administrative interface and APIs provided by the application.

1. Use the application console or APIs to connect the application to your instance of IAM Identity Center

Use the console for the AWS managed application or the application APIs to connect the application to your instance of IAM Identity Center. When you use the console for the application, the administrative user interface includes a widget that streamlines the setup and connection process.

2. Use the application console or APIs to set up user access to the application's resources

Complete this step to authorize which resources, or data, a user can access. Access is based on the user's identity or group membership. The authorization model varies based on the application.

Important

You must complete this step to enable users to access the AWS service's resources. Otherwise, users can't access resources, even if the requesting application is authorized to request access to the service.

Trusted identity propagation request flows for AWS managed applications

All trusted identity propagation flows to AWS managed applications must begin with an application that obtains a token from IAM Identity Center. This token is required because it contains a reference to a user that is known to IAM Identity Center and applications that are registered with IAM Identity Center.

The following sections describe the ways in which an AWS managed application can obtain a token from IAM Identity Center to initiate trusted identity propagation.

Topics

- [Web-based, IAM Identity Center authentication](#)
- [Console-based, user-initiated authentication requests](#)

Web-based, IAM Identity Center authentication

For this flow, the AWS managed application provides a web-based single sign-on experience using IAM Identity Center for authentication.

When a user opens an AWS managed application, a single sign-on flow that uses IAM Identity Center is triggered. If there isn't an active session for the user in IAM Identity Center, the user is presented with a sign-in page based on the identity source that you have specified, and IAM Identity Center creates a session for the user.

IAM Identity Center provides the AWS managed application with a token that includes the user's identity and a list of audiences (Auds) and related scopes that the application is registered to use. The application can then use the token to make requests to other receiving AWS services.

Console-based, user-initiated authentication requests

For this flow, the AWS managed application provides a console experience that users initiate.

In this case, the AWS managed application is entered from the AWS Management Console after assuming a role. For the application to obtain a token, the user must initiate a process to trigger the application to authenticate the user. This initiates authentication using IAM Identity Center, which will redirect the user to the identity source that you have configured.

After an application obtains a token

After a requesting application obtains a token from IAM Identity Center, the application periodically refreshes the token, which can be used for the life of the user's session. During this time, the application might:

- Obtain more information about the token to determine who the user is and which scopes the application can use with other receiving AWS managed applications.
- Pass the token in calls to other receiving AWS managed applications that support the use of tokens.
- Obtain identity-enhanced IAM role sessions that it can use to make requests to other AWS managed applications that use AWS Signature Version 4.

An identity-enhanced IAM role session is an IAM role session that contains the user's propagated identity stored in a token that is created by IAM Identity Center.

Identity-enhanced IAM role sessions

The AWS Security Token Service enables an AWS managed application or a customer managed application to obtain an identity-enhanced IAM role session. The application can use this identity-enhanced role session when making requests to an AWS managed application. These identity-

enhanced role sessions have an added identity context that the requested AWS managed application can use to authorize access.

An application obtains an identity-enhanced role session by making a request to the AWS STS [AssumeRole](#) API action and passing a context assertion in the `ProvidedContexts` parameter of the `AssumeRole` request. The context assertion is obtained from the `idToken` claim that is available in the response from the SSO OIDC [CreateTokenWithIAM](#) request. This context adds the `userId` of the IAM Identity Center user to the identity-enhanced role session context.

When an AWS managed application uses an identity-enhanced role session to access a resource, CloudTrail logs the user's identity (`userId`), the initiating session, and the action taken. For more information, see [Identity-enhanced IAM role session logging](#).

Types of identity-enhanced IAM role sessions

AWS STS can create two different types of identity-enhanced IAM role sessions, depending on the context assertion provided to the `AssumeRole` request:

- AWS managed applications that receive a role session with `sts:audit_context` only log the user's identity (`userId`) to CloudTrail.
- AWS managed applications that receive a role session with `sts:identity_context` can authorize access based on the propagated user identity and logs the `userId` to CloudTrail.

An identity-enhanced role session can have only one of these context assertions, not both.

Identity-enhanced IAM role sessions created with `sts:audit_context`

AWS managed applications and customer managed applications that only require the IAM Identity Center user's `userId` to be logged with actions taken by the IAM role, must obtain an identity-enhanced IAM role session with `sts:audit_context`. This context assertion is compatible with all AWS API actions that the role has permission to call.

Note

When using `sts:audit_context` in the `ProvidedContexts` parameter of the [AssumeRole](#) request, the called AWS service uses only the role to authorize access. These sessions can't be used for authorizing access based on the IAM Identity Center user.

To obtain this type of identity-enhanced role session from AWS STS, AWS managed applications and customer managed applications provide the value of the `sts:audit_context` field in the [AssumeRole](#) request using the `ProvidedContexts` request parameter. Use `arn:aws:iam::aws:contextProvider/IdentityCenter` as the value for `ProviderArn`.

Identity-enhanced IAM role sessions created with `sts:identity_context`

When an AWS managed application or customer managed application requires the requested AWS service to authorize access based on the user's identity or group membership, the application must use an identity-enhanced IAM role session that is created with `sts:identity_context`. In addition to the requested AWS managed application authorizing access based on the user, CloudTrail logs the `userId` of the IAM Identity Center user with all actions taken by the role.

To obtain this type of identity-enhanced role session from AWS STS, AWS managed applications and customer managed applications provide the value of the `sts:identity_context` field in the [AssumeRole](#) request using the `ProvidedContexts` request parameter. Use `arn:aws:iam::aws:contextProvider/IdentityCenter` as the value for `ProviderArn`.

Important

If an AWS managed application or customer managed application makes a request to an AWS service using an identity-enhanced IAM role session that is created with `sts:identity_context`, and the API action that is called is not enabled for trusted identity propagation, the request will result in an error. The [AWSIAMIdentityCenterAllowListForIdentityContext](#) AWS managed policy, which is supplied as a session policy, lists supported AWS service API actions that can be used with `sts:identity_context`. This policy prevents the use of unsupported AWS services. To view the permissions for this policy, see [AWSIAMIdentityCenterAllowListForIdentityContext](#) in the *AWS Managed Policy Reference*.

An AWS managed application or customer managed application that makes requests using `sts:identity_context` will also receive an error if the called AWS managed application is not connected to IAM Identity Center or the requested resource is not configured for user or group-membership based authorization.

To avoid this issue, do either of the following:

- Verify that the receiving AWS managed application is connected to IAM Identity Center.

- Use the console for the receiving AWS managed application or its APIs to configure authorization to its resources based on the user's identity or group memberships. The setup requirements for this vary based on the AWS managed application.

For more information, see the documentation for the receiving AWS managed application.

Identity-enhanced IAM role session logging

When a request is made to an AWS service using an identity-enhanced IAM role session that is created with `sts:audit_context` or the `sts:identity_context`, the user's IAM Identity Center `userId` is logged to CloudTrail in the `OnBehalfOf` element. The way in which events are logged in CloudTrail varies based on the AWS service. Not all AWS services log the `onBehalfOf` element.

The following is an example of how a request made to an AWS service using an identity-enhanced role session is logged in CloudTrail.

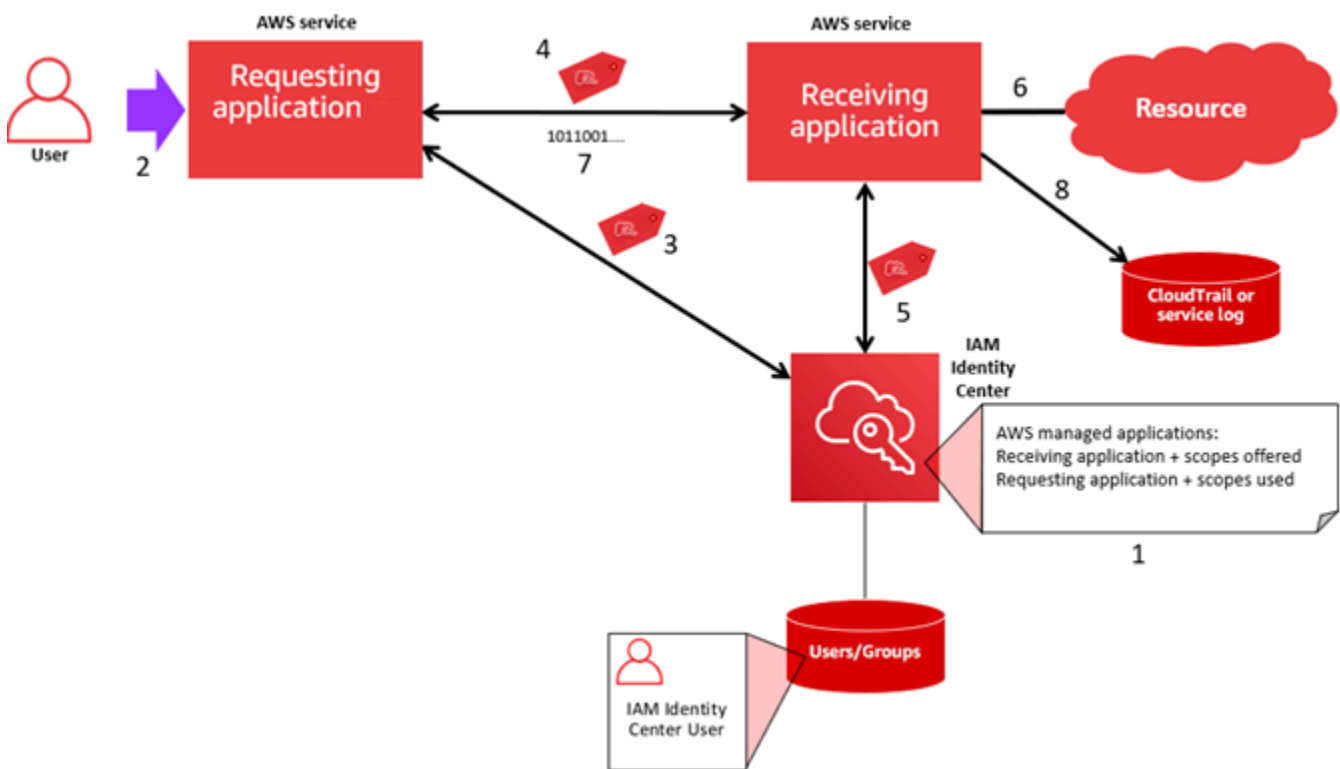
```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROEXAMPLE:MyRole",
  "arn": "arn:aws:sts::111111111111:assumed-role/MyRole/MySession",
  "accountId": "111111111111",
  "accessKeyId": "ASIAEXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROEXAMPLE",
      "arn": "arn:aws:iam::111111111111:role/MyRole",
      "accountId": "111111111111",
      "userName": "MyRole"
    },
    "attributes": {
      "creationDate": "2023-12-12T13:55:22Z",
      "mfaAuthenticated": "false"
    }
  },
  "onBehalfOf": {
    "userId": "11111111-1111-1111-1111-111111111111",
    "identityStoreArn": "arn:aws:identitystore::111111111111:identitystore/d-1111111111"
  }
}
```

}

Setup process and request flow for AWS managed applications

This section describes the setup process and request flow for AWS managed applications that use trusted identity propagation and that provide a web-based single sign-on experience.

The following diagram provides an overview of this process.



The following steps provide additional information about this process.

1. Use the console for the AWS managed application or the application APIs to do the following:
 - a. Connect the application to your instance of IAM Identity Center.
 - b. Set up permissions to authorize which application resources a user can access.
2. The request flow begins when a user opens an AWS managed application that can request access to resources (a requesting application).
3. To obtain a token to access the receiving AWS managed application, the requesting AWS managed application initiates a sign-in request to IAM Identity Center.

If the user isn't signed in, IAM Identity Center triggers a user authentication flow to the identity source that you have specified. This creates a new AWS access portal session for the user with

the duration that you configured in IAM Identity Center. IAM Identity Center then generates a token that is associated with the session, and the application can operate for the remaining duration of the user's AWS access portal session. If the user signs out of their application, or if you delete their session, the session automatically ends within two hours.

4. The AWS managed application initiates a request to the receiving application and provides its token.
5. The receiving application makes calls to IAM Identity Center to obtain the identity of the user and the scopes that are encoded in the token. The receiving application might also make requests to obtain user attributes or the user's group memberships from the Identity Center directory.
6. The receiving application uses its authorization configuration to determine if the user is authorized to access the requested application resource.
7. If the user is authorized to access the requested application resource, the receiving application responds to the request.
8. The user's identity, actions performed on their behalf, and other events recorded in the receiving application logs and AWS CloudTrail events. The specific way in which this information is logged varies based on the application.

Using trusted identity propagation with customer managed applications

Trusted identity propagation enables a customer managed application to request access to data in AWS services on behalf of a user. Data access management is based on a user's identity, so administrators can grant access based on users' existing user and group memberships. The user's identity, actions performed on their behalf, and other events are recorded in service-specific logs and CloudTrail events.

With trusted identity propagation, a user can sign in to a customer managed application, and that application can pass the user's identity in requests to access data in AWS services.

Important

To access an AWS service, customer managed applications must obtain a token from a trusted token issuer, which is external to IAM Identity Center. A *trusted token issuer* is an OAuth 2.0 authorization server that creates signed tokens. These tokens authorize applications that initiate requests for access to AWS services (receiving applications). For more information, see [Using applications with a trusted token issuer](#).

Topics

- [Set up customer managed OAuth 2.0 applications for trusted identity propagation](#)
- [Specify trusted applications](#)

Set up customer managed OAuth 2.0 applications for trusted identity propagation

To set up a customer managed OAuth 2.0 application for trusted identity propagation, you must first add it to IAM Identity Center. Use the following procedure to add your application to IAM Identity Center.

Topics

- [Step 1: Select application type](#)
- [Step 2: Specify application details](#)
- [Step 3: Specify authentication settings](#)
- [Step 4: Specify application credentials](#)
- [Step 5: Review and configure](#)

Step 1: Select application type

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **Customer managed** tab.
4. Choose **Add application**.
5. On the **Select application type** page, under **Setup preference**, choose **I have an application I want to set up**.
6. Under **Application type**, choose **OAuth 2.0**.
7. Choose **Next** to proceed to the next page, [Step 2: Specify application details](#).

Step 2: Specify application details

1. On the **Specify application details** page, under **Application name and description**, enter a **Display name** for the application, such as **MyApp**. Then, enter a **Description**.
2. Under **User and group assignment method**, choose one of the following options:

- **Require assignments** – Allow only IAM Identity Center users and groups who are assigned to this application to access the application.

Application tile visibility – Only users who are assigned to the application directly or through a group assignment can view the application tile in the AWS access portal, provided that **Application visibility in AWS access portal** is set to **Visible**.

- **Do not require assignments** – Allow all authorized IAM Identity Center users and groups to access this application.

Application tile visibility – The application tile is visible to all users who sign in to the AWS access portal, unless **Application visibility in AWS access portal** is set to **Not visible**.

3. Under **AWS access portal**, enter the URL where users can access the application and specify whether the application tile will be visible or not visible in the AWS access portal. If you choose **Not visible**, not even assigned users can view the application tile.
4. Under **Tags (optional)**, choose **Add new tag**, and then specify values for **Key** and **Value (optional)**.

For information about tags, see [Tagging AWS IAM Identity Center resources](#).

5. Choose **Next**, and proceed to the next page, [Step 3: Specify authentication settings](#).

Step 3: Specify authentication settings

To add a customer managed application that supports OAuth 2.0 to IAM Identity Center, you must specify a trusted token issuer. A trusted token issuer is an OAuth 2.0 authorization server that creates signed tokens. These tokens authorize applications that initiate requests (requesting applications) for access to AWS managed applications (receiving applications).

1. On the **Specify authentication settings** page, under **Trusted token issuers**, do either of the following:
 - To use an existing trusted token issuer:

Select the check box next to the name of the trusted token issuer that you want to use.
 - To add a new trusted token issuer:
 1. Choose **Create trusted token issuer**.
 2. A new browser tab opens. Follow steps 5 through 8 in [How to add a trusted token issuer to the IAM Identity Center console](#).

3. After you complete these steps, return to the browser window that you are using for your application setup and select the trusted token issuer that you just added.
4. In the list of trusted token issuers, select the check box next to the name of the trusted token issuer that you just added.

After you select a trusted token issuer, the **Configure selected trusted token issuers** section appears.

2. Under **Configure selected trusted token issuers**, enter the **Aud claim**. The **Aud claim** identifies the intended audience (recipients) for the token that is generated by the trusted token issuer. For more information, see [Aud claim](#).
3. To prevent your users from having to reauthenticate when they are using this application, select **Automatically refresh user authentication for active application session**. When selected, this option refreshes the access token for the session every 60 minutes, until the session expires or the user ends the session.
4. Choose **Next**, and proceed to the next page, [Step 4: Specify application credentials](#).

Step 4: Specify application credentials

Complete the steps in this procedure to specify the credentials that your application uses to perform token exchange actions with trusted applications. These credentials are used in a resource-based policy. The policy requires that you specify a principal that has permissions to perform the actions that are specified in the policy. **You must specify a principal**, even if the trusted applications are in the same AWS account.

Note

When you set permissions with policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as least-privilege permissions.

This policy requires the `sso-oauth:CreateTokenWithIAM` action.

1. On the **Specify application credentials** page, do either of the following:
 - To quickly specify one or more IAM roles:
 1. Choose **Enter one or more IAM roles**.

2. Under **Enter IAM roles**, specify the Amazon Resource Name (ARN) of an existing IAM role. To specify the ARN, use the following syntax. The Region portion of the ARN is blank because IAM resources are global.

```
arn:aws:iam::account:role/role-name-with-path
```

For more information, see [Cross-account access using resource-based policies](#) and [IAM ARNs](#) in the *AWS Identity and Access Management User Guide*.

- To manually edit the policy (required if you specify non-AWS credentials):
 1. Select **Edit the application policy**.
 2. Modify your policy by typing or pasting text in the JSON text box.
 3. Resolve any security warnings, errors, or general warnings generated during policy validation. For more information see [Validating IAM policies](#) in the *AWS Identity and Access Management User Guide*.
2. Choose **Next** and proceed to the next page, [Step 5: Review and configure](#).

Step 5: Review and configure

1. On the **Review and configure** page, review the choices that you made. To make changes, choose the configuration section that you want, choose **Edit**, and then make the required changes.
2. After you're finished, choose **Add application**.
3. The application that you added appears in the **Customer managed applications** list.
4. After you set up your customer managed application in IAM Identity Center, you must specify one or more AWS services, or trusted applications, for identity propagation. This enables users to sign in to your customer managed application and access data in the trusted application.

For more information, see [Specify trusted applications](#) .

Specify trusted applications

After you [set up your customer managed application](#), you must specify one or more trusted AWS services, or trusted applications, for identity propagation. Specify an AWS service that has data that users of your customer managed applications need to access. When your users sign in to

your customer managed application, that application will pass your users' identity to the trusted application.

Use the following procedure to select a service, and then specify individual applications to trust for that service.

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. Choose the **Customer managed** tab.
4. In the **Customer managed applications** list, select the OAuth 2.0 application that you want to initiate requests for access. This is the application that your users sign in to.
5. On the **Details page**, under **Trusted applications for identity propagation**, choose **Specify trusted applications**.
6. Under **Setup type**, select **Individual applications and specify access**, and then choose **Next**.
7. On the **Select service** page, choose the AWS service that has applications that your customer managed application can trust for identity propagation, and then choose **Next**.

The service that you select defines the applications that can be trusted. You'll select applications in the next step.

8. On the **Select applications** page, choose **Individual applications**, select the check box for each application that can receive requests for access, and then choose **Next**.
9. On the **Configure access** page, under **Configuration method**, do either of the following:
 - **Select access per application** – Select this option to configure different access levels for each application. Choose the application for which you want to configure the access level, and then choose **Edit access**. In **Level of access to apply**, change the access levels as needed, and then choose **Save changes**.
 - **Apply same level of access to all applications** – Select this option if you don't need to configure access levels on a per-application basis.
10. Choose **Next**.
11. On the **Review configuration** page, review the choices that you made. To make changes, choose the configuration section that you want, choose **Edit access**, and then make the required changes.
12. After you're finished, choose **Trust applications**.

Using applications with a trusted token issuer

Trusted token issuers enable you to use trusted identity propagation with applications that authenticate outside of AWS. With trusted token issuers, you can authorize these applications to make requests on behalf of their users to access AWS managed applications.

The following topics describe how trusted token issuers work and provide setup guidance.

Topics

- [Trusted token issuer overview](#)
- [Prerequisites and considerations for trusted token issuers](#)
- [JTI claim details](#)
- [Trusted token issuer configuration settings](#)
- [Setting up a trusted token issuer](#)

Trusted token issuer overview

Trusted identity propagation provides a mechanism that enables applications that authenticate outside of AWS to make requests on behalf of their users with the use of a trusted token issuer. A *trusted token issuer* is an OAuth 2.0 authorization server that creates signed tokens. These tokens authorize applications that initiate requests (requesting applications) for access to AWS services (receiving applications). Requesting applications initiate access requests on behalf of users that the trusted token issuer authenticates. The users are known to both the trusted token issuer and IAM Identity Center.

AWS services that receive requests manage fine-grained authorization to their resources based on their users and group membership as represented in the Identity Center directory. AWS services can't use the tokens from the external token issuer directly.

To solve this, IAM Identity Center provides a way for the requesting application, or an AWS driver that the requesting application uses, to exchange the token issued by the trusted token issuer for a token that is generated by IAM Identity Center. The token that is generated by IAM Identity Center refers to the corresponding IAM Identity Center user. The requesting application, or the driver, uses the new token to initiate a request to the receiving application. Because the new token references the corresponding user in IAM Identity Center, the receiving application can authorize the requested access based on the user or their group membership as represented in IAM Identity Center.

⚠ Important

Choosing an OAuth 2.0 authorization server to add as a trusted token issuer is a security decision that requires careful consideration. Only choose trusted token issuers that you trust to perform the following tasks:

- Authenticate the user who is specified in the token.
- Authorize that user's access to the receiving application.
- Generate a token that IAM Identity Center can exchange for an IAM Identity Center created token.

Prerequisites and considerations for trusted token issuers

Before you set up a trusted token issuer, review the following prerequisites and considerations.

• **Trusted token issuer configuration**

You must configure an OAuth 2.0 authorization server (the trusted token issuer). Although the trusted token issuer is typically the identity provider that you use as your identity source for IAM Identity Center, it doesn't have to be. For information about how to set up the trusted token issuer, see the documentation for the relevant identity provider.

ℹ Note

You can configure up to 10 trusted token issuers for use with IAM Identity Center, as long you map the identity of each user in the trusted token issuer to a corresponding user in IAM Identity Center.

- The OAuth 2.0 authorization server (the trusted token issuer) that creates the token must have an [OpenID Connect \(OIDC\)](#) discovery endpoint that IAM Identity Center can use to obtain public keys to verify the token signatures. For more information, see [OIDC discovery endpoint URL \(issuer URL\)](#).

• **Tokens issued by the trusted token issuer**

Tokens from the trusted token issuer must meet the following requirements:

- The token must be signed and in [JSON Web Token \(JWT\)](#) format using the RS256 algorithm.
- The token must contain the following claims:

- [Issuer](#) (iss) – The entity that issued the token. This value must match the value that is configured in the OIDC discovery endpoint (issuer URL) in the trusted token issuer.
- [Subject](#) (sub) – The authenticated user.
- [Audience](#) (aud) – The intended recipient of the token. This is the AWS service that will be accessed after the token is exchanged for a token from IAM Identity Center. For more information, see [Aud claim](#).
- [Expiration Time](#) (exp) – The time after which the token expires.
- The token can be an identity token or an access token.
- The token must have an attribute that can be mapped uniquely to one IAM Identity Center user.
- **Optional claims**

IAM Identity Center supports all optional claims that are defined in RFC 7523. For more information, see [Section 3: JWT Format and Processing Requirements](#) of this RFC.

For example, the token can contain a [JTI \(JWT ID\) claim](#). This claim, when present, prevents tokens that have the same JTI from being reused for token exchanges. For more information about JTI claims, see [JTI claim details](#).

- **IAM Identity Center configuration to work with a trusted token issuer**

You must also enable IAM Identity Center, configure the identity source for IAM Identity Center, and provision users that correspond to the users in the trusted token issuer's directory.

To do this, you must do either of the following:

- Synchronize users into IAM Identity Center by using the System for Cross-domain Identity Management (SCIM) 2.0 protocol.
- Create the users directly in IAM Identity Center.

 **Note**

Trusted token issuers aren't supported if you use Active Directory Domain Service as your identity source.

JTI claim details

If IAM Identity Center receives a request to exchange a token that IAM Identity Center has already exchanged, the request fails. To detect and prevent reuse of a token for token exchanges, you can include a JTI claim. IAM Identity Center protects against the replay of tokens based on the claims in the token.

Not all OAuth 2.0 authorization servers add a JTI claim to tokens. Some OAuth 2.0 authorization servers might not allow you to add a JTI as a custom claim. OAuth 2.0 authorization servers that support the use of a JTI claim might add this claim to identity tokens only, access tokens only, or both. For more information, see the documentation for your OAuth 2.0 authorization server.

For information about building applications that exchange tokens, see the IAM Identity Center API documentation. For information about configuring a customer managed application to obtain and exchange the correct tokens, see the documentation for the application.

Trusted token issuer configuration settings

The following sections describe the settings required to set up and use a trusted token issuer.

Topics

- [OIDC discovery endpoint URL \(issuer URL\)](#)
- [Attribute mapping](#)
- [Aud claim](#)

OIDC discovery endpoint URL (issuer URL)

When you add a trusted token issuer to the IAM Identity Center console, you must specify the OIDC discovery endpoint URL. This URL is commonly referred to by its relative URL, `/.well-known/openid-configuration`. In the IAM Identity Center console, this URL is called the *issuer URL*.

Note

You must paste the URL of the discovery endpoint *up until and without* `/.well-known/openid-configuration`. If `/.well-known/openid-configuration` is included in the URL, the trusted token issuer configuration won't work. Because IAM Identity Center doesn't validate this URL, if the URL isn't correctly formed, the trusted token issuer setup will fail without notification.

The OIDC discovery endpoint URL must be reachable via ports 80 and 430 only.

IAM Identity Center uses this URL to obtain additional information about the trusted token issuer. For example, IAM Identity Center uses this URL to obtain the information required to verify the tokens that the trusted token issuer generates. When you add a trusted token issuer to IAM Identity Center, you must specify this URL. To find the URL, see the documentation for the OAuth 2.0 authorization server provider that you use to generate tokens for your application, or contact the provider directly for assistance.

Attribute mapping

Attribute mappings enable IAM Identity Center to match the user that is represented in a token issued by a trusted token issuer to a single user in IAM Identity Center. You must specify the attribute mapping when you add the trusted token issuer to IAM Identity Center. This attribute mapping is used in a claim in the token that is generated by the trusted token issuer. The value in the claim is used to search IAM Identity Center. The search uses the specified attribute to retrieve a single user in IAM Identity Center, who will be used as the user within AWS. The claim that you choose must be mapped to one attribute in a fixed list of available attributes in the IAM Identity Center identity store. You can choose one of the following IAM Identity Center identity store attributes: user name, email, and external ID. The value for the attribute that you specify in IAM Identity Center must be unique for each user.

Aud claim

An *aud claim* identifies the audience (recipients) for which a token is intended. When the application requesting access authenticates through an identity provider that is not federated to IAM Identity Center, that identity provider must be set up as a trusted token issuer. The application that receives the access request (the receiving application) must exchange the token that is generated by the trusted token issuer for a token that is generated by IAM Identity Center.

For information about how to obtain the aud claim values for the receiving application as they are registered in the trusted token issuer, see the documentation for your trusted token issuer or contact the trusted token issuer administrator for assistance.

Setting up a trusted token issuer

To enable trusted identity propagation for an application that authenticates externally to IAM Identity Center, one or more administrators must set up a trusted token issuer. A trusted token issuer is an OAuth 2.0 authorization server that issues tokens to applications that initiate requests

(requesting applications). The tokens authorize these applications to initiate requests on behalf of their users to a receiving application (an AWS service).

Topics

- [Coordinating administrative roles and responsibilities](#)
- [Tasks for setting up a trusted token issuer](#)
- [How to add a trusted token issuer to the IAM Identity Center console](#)
- [How to view or edit trusted token issuer settings in the IAM Identity Center console](#)
- [Setup process and request flow for applications that use a trusted token issuer](#)

Coordinating administrative roles and responsibilities

In some cases, a single administrator might perform all of the necessary tasks for setting up a trusted token issuer. If multiple administrators perform these tasks, close coordination is required. The following table describes how multiple administrators might coordinate to set up a trusted token issuer and configure AWS service to use it.

Note

The application can be any AWS service that is integrated with IAM Identity Center and supports trusted identity propagation.

For more information, see [Tasks for setting up a trusted token issuer](#).

Role	Performs these tasks	Coordinates with
IAM Identity Center administrator	<p>Adds the external IdP as a trusted token issuer to the IAM Identity Center console.</p> <p>Helps set up the correct attribute mapping between IAM Identity Center and the external IdP.</p> <p>Notifies the AWS service administrator when the trusted token issuer</p>	<p>External IdP (trusted token issuer) administrator</p> <p>AWS service administrator</p>

Role	Performs these tasks	Coordinates with
	is added to the IAM Identity Center console.	
External IdP (trusted token issuer) administrator	<p>Configures the external IdP to issue tokens.</p> <p>Helps set up the correct attribute mapping between IAM Identity Center and the external IdP.</p> <p>Provides the audience name (Aud claim) to the AWS service administrator.</p>	<p>IAM Identity Center administrator</p> <p>AWS service administrator</p>
AWS service administrator	<p>Checks the AWS service console for the trusted token issuer. The trusted token issuer will be visible in the AWS service console after the IAM Identity Center administrator adds it to the IAM Identity Center console.</p> <p>Configures the AWS service to use the trusted token issuer.</p>	<p>IAM Identity Center administrator</p> <p>External IdP (trusted token issuer) administrator</p>

Tasks for setting up a trusted token issuer

To set up a trusted token issuer, an IAM Identity Center administrator, external IdP (trusted token issuer) administrator, and application administrator must complete the following tasks.

Note

The application can be any AWS service that is integrated with IAM Identity Center and supports trusted identity propagation.

1. **Add the trusted token issuer to IAM Identity Center** – The IAM Identity Center administrator [adds the trusted token issuer by using the IAM Identity Center console](#) or APIs. This configuration requires specifying the following:
 - A name for the trusted token issuer.
 - The OIDC discovery endpoint URL (in the IAM Identity Center console, this URL is called the *issuer URL*). The discovery endpoint must be reachable via ports 80 and 443 only.
 - Attribute mapping for user lookup. This attribute mapping is used in a claim in the token that is generated by the trusted token issuer. The value in the claim is used to search IAM Identity Center. The search uses the specified attribute to retrieve a single user in IAM Identity Center.
2. **Connect the AWS service to IAM Identity Center** – The AWS service administrator must connect the application to IAM Identity Center by using the console for the application or the application APIs.

After the trusted token issuer is added to the IAM Identity Center console, it is also visible in the AWS service console and available for the AWS service administrator to select.

3. **Configure the use of token exchange** – In the AWS service console, the AWS service administrator configures AWS service to accept tokens issued by the trusted token issuer. These tokens are exchanged for tokens generated by IAM Identity Center. This requires specifying the name of the trusted token issuer from Step 1, and the Aud claim value that corresponds to the AWS service.

The trusted token issuer places the Aud claim value in the token it issues to indicate that the token is intended for use by the AWS service. To obtain this value, contact the administrator for the trusted token issuer.


How to add a trusted token issuer to the IAM Identity Center console

In an organization that has multiple administrators, this task is performed by an IAM Identity Center administrator. If you are the IAM Identity Center administrator, you must choose which external IdP to use as a trusted token issuer.

To add a trusted token issuer to the IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.

3. On the **Settings** page, choose the **Authentication** tab.
4. Under **Trusted token issuers**, choose **Create trusted token issuer**.
5. On the **Set up an external IdP to issue trusted tokens** page, under **Trusted token issuer details**, do the following:
 - For **Issuer URL**, specify the OIDC discovery URL of the external IdP that will issue tokens for trusted identity propagation. You must specify the URL of the discovery endpoint up until and without `.well-known/openid-configuration`. The administrator of the external IdP can provide this URL.

 **Note**

Note This URL must match the URL in the Issuer (iss) claim in tokens that are issued for trusted identity propagation.

- For **Trusted token issuer name**, enter a name to identify this trusted token issuer in IAM Identity Center and in the application console.
6. Under **Map attributes**, do the following:
 - For **Identity provider attribute**, select an attribute from the list to map to an attribute in the IAM Identity Center identity store.
 - For **IAM Identity Center attribute**, select the corresponding attribute for the attribute mapping.
 7. Under **Tags (optional)**, choose **Add new tag**, specify a value for **Key**, and optionally for **Value**.

For information about tags, see [Tagging AWS IAM Identity Center resources](#).

8. Choose **Create trusted token issuer**.
9. After you finish creating the trusted token issuer, contact the application administrator to let them know the name of the trusted token issuer, so that they can confirm that the trusted token issuer is visible in the applicable console.
10. The application administrator must select this trusted token issuer in the applicable console to enable user access to the application from applications that are configured for trusted identity propagation.

How to view or edit trusted token issuer settings in the IAM Identity Center console

After you add a trusted token issuer to the IAM Identity Center console, you can view and edit the relevant settings.

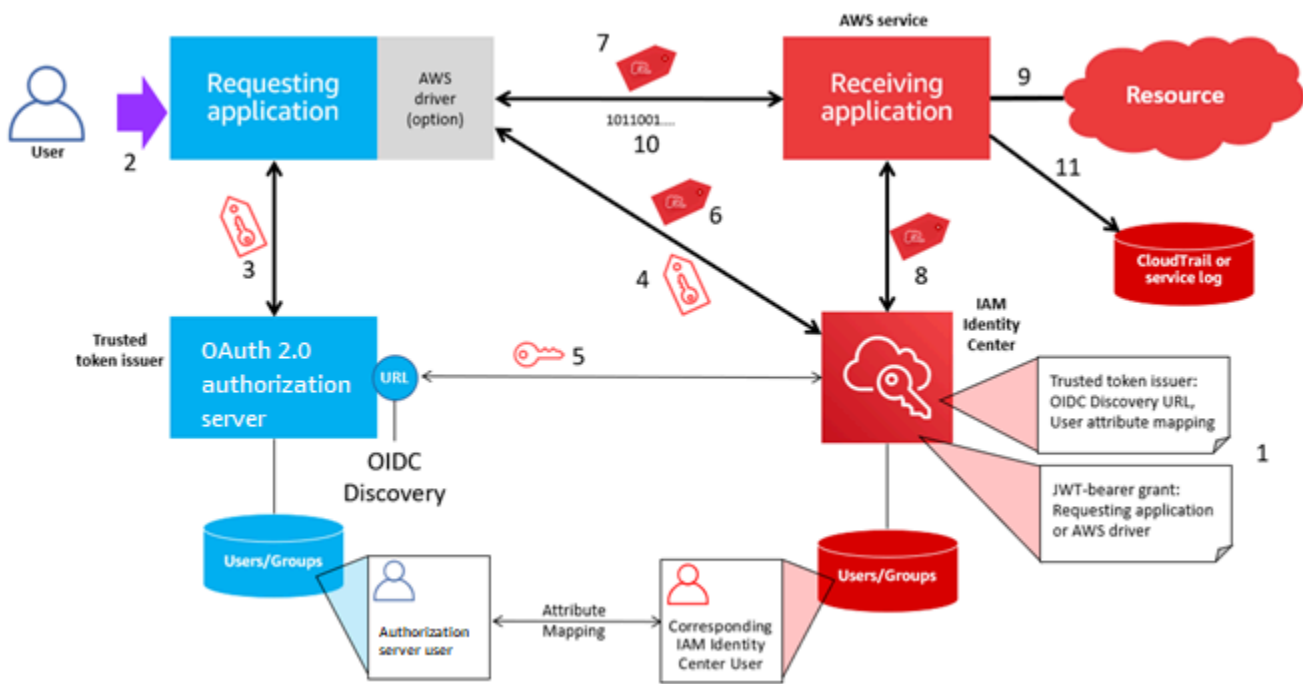
If you plan to edit the trusted token issuer settings, keep in mind that doing so might cause users to lose access to any applications that are configured to use the trusted token issuer. To avoid disrupting user access, we recommend that you coordinate with the administrators for any applications that are configured to use the trusted token issuer before you edit settings.

To view or edit trusted token issuer settings in the IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Authentication** tab.
4. Under **Trusted token issuers**, select the trusted token issuer that you want to view or edit.
5. Choose **Actions**, and then choose **Edit**.
6. On the **Edit trusted token issuer** page, view or edit settings as needed. You can edit the trusted token issuer name, attribute mappings, and tags.
7. Choose **Save changes**.
8. In the **Edit trusted token issuer** dialog box, you're prompted to confirm that you want to make changes. Choose **Confirm**.

Setup process and request flow for applications that use a trusted token issuer


This section describes the setup process and request flow for applications that use a trusted token issuer for trusted identity propagation. The following diagram provides an overview of this process.



The following steps provide additional information about this process.

1. Set up IAM Identity Center and the receiving AWS managed application to use a trusted token issuer. For information, see [Tasks for setting up a trusted token issuer](#).
2. The request flow begins when a user opens the requesting application.
3. The requesting application requests a token from the trusted token issuer to initiate requests to the receiving AWS managed application. If the user hasn't authenticated yet, this process triggers an authentication flow. The token contains the following information:
 - The subject (Sub) of the user.
 - The attribute that IAM Identity Center uses to look up the corresponding user in IAM Identity Center.
 - An audience (Aud) claim that contains a value that the trusted token issuer associates with the receiving AWS managed application. If other claims are present, they aren't used by IAM Identity Center.
4. The requesting application, or the AWS driver that it uses, passes the token to IAM Identity Center and requests that the token be exchanged for a token that is generated by IAM Identity Center. If you use an AWS driver, you might need to configure the driver for this use case. For more information, see the documentation for the relevant AWS managed application.
5. IAM Identity Center uses the OIDC Discovery endpoint to obtain the public key that it can use to verify the authenticity of the token. IAM Identity Center then does the following:

- Verifies the token.
- Searches the Identity Center directory. To do this, IAM Identity Center uses the mapped attribute specified in the token.
- Verifies that the user is authorized to access the receiving application. If the AWS managed application is configured to require assignments to users and groups, the user must have a direct or group-based assignment to the application; otherwise the request is denied. If the AWS managed application is configured to not require user and group assignments, processing continues.

 **Note**

AWS services have a default setting configuration that determines whether assignments are required for users and groups. We recommend that you do not modify the **Require assignments** setting for these applications if you plan to use them with trusted identity propagation. Even if you have configured fine-grained permissions that allow user access to specific application resources, modifying the **Require assignments** setting might result in unexpected behavior, including disrupted user access to these resources.

- Verifies that the requesting application is configured to use valid scopes for the receiving AWS managed application.
6. If the previous verification steps are successful, IAM Identity Center creates a new token. The new token is an opaque (encrypted) token that includes the identity of the corresponding user in IAM Identity Center, the audience (Aud) of the receiving AWS managed application, and the scopes that the requesting application can use when making requests to the receiving AWS managed application.
 7. The requesting application, or the driver that it uses, initiates a resource request to the receiving application and passes the token that IAM Identity Center generated to the receiving application.
 8. The receiving application makes calls to IAM Identity Center to obtain the identity of the user and the scopes that are encoded in the token. It might also make requests to obtain user attributes or the user's group memberships from the Identity Center directory.
 9. The receiving application uses its authorization configuration to determine if the user is authorized to access the requested application resource.

10. If the user is authorized to access the requested application resource, the receiving application responds to the request.
11. The user's identity, actions performed on their behalf, and other events recorded in the receiving application logs and CloudTrail events. The specific way in which this information is logged varies based on the application.

Manage IAM Identity Center certificates

IAM Identity Center uses certificates to set up a SAML trust relationship between IAM Identity Center and your application's service provider. When you add an application in IAM Identity Center, an IAM Identity Center certificate is automatically created for use with that application during the setup process. By default, this autogenerated IAM Identity Center certificate is valid for a period of five years.

As an IAM Identity Center administrator, you'll occasionally need to replace older certificates with newer ones for a given application. For example, you might need to replace a certificate when the expiration date on the certificate approaches. The process of replacing an older certificate with a newer one is referred to as *certificate rotation*.

Topics

- [Considerations before rotating a certificate](#)
- [Rotate an IAM Identity Center certificate](#)
- [Certificate expiration status indicators](#)

Considerations before rotating a certificate

Before you start the process of rotating a certificate in IAM Identity Center, consider the following:

- The certification rotation process requires that you reestablish the trust between IAM Identity Center and the service provider. To reestablish the trust, use the procedures provided in [Rotate an IAM Identity Center certificate](#).
- Updating the certificate with the service provider may cause a temporary service disruption for your users until the trust has been successfully reestablished. Plan this operation carefully during off peak hours if possible.

Rotate an IAM Identity Center certificate

Rotating an IAM Identity Center certificate is a multistep process that involves the following:

- Generating a new certificate
- Adding the new certificate to the service provider's website
- Setting the new certificate to active
- Deleting the inactive certificate

Use all of the following procedures in the following order to complete the certificate rotation process for a given application.

Step 1: Generate a new certificate.

New IAM Identity Center certificates that you generate can be configured to use the following properties:

- **Validity period** – Specifies the time allotted (in months) before a new IAM Identity Center certificate expires.
- **Key size** – Determines the number of bits that a key must use with its cryptographic algorithm. You can set this value to either 1024-bit RSA or 2048-bit RSA. For general information about how key sizes work in cryptography, see [Key size](#).
- **Algorithm** – Specifies the algorithm that IAM Identity Center uses when signing the SAML assertion/response. You can set this value to either SHA-1 or SHA-256. AWS recommends using SHA-256 when possible, unless your service provider requires SHA-1. For general information about how cryptography algorithms work, see [Public-key cryptography](#).

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, choose the application that you want to generate a new certificate for.
4. On the application details page, choose the **Configuration** tab. Under **IAM Identity Center metadata**, choose **Manage certificate**. If you don't have a **Configuration** tab or the configuration setting is not available, you do not need to rotate the certificate for this application.
5. On the **IAM Identity Center certificate** page, choose **Generate new certificate**.

6. In the **Generate new IAM Identity Center certificate** dialog box, specify the appropriate values for **Validity period**, **Algorithm**, and **Key size**. Then choose **Generate**.

Step 2: Update the service provider's website.

Use the following procedure to reestablish the trust with the application's service provider.

Important

When you upload the new certificate to the service provider, your users might not be able to get authenticated. To correct this situation, set the new certificate as active as described in the next step.

1. In the [IAM Identity Center console](#), choose the application that you just generated a new certificate for.
2. On the application details page, choose **Edit configuration**.
3. Choose **View instructions**, and then follow the instructions for your specific application service provider's website to add the newly generated certificate.

Step 3: Set the new certificate to active.

An application can have up to two certificates assigned to it. IAM Identity Center will use the certification that is set as active to sign all SAML assertions.

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, choose your application.
4. On the application details page, choose the **Configuration** tab. Under **IAM Identity Center metadata**, choose **Manage certificate**.
5. On the **IAM Identity Center certificate** page, select the certificate you want to set to active, choose **Actions**, and then choose **Set as active**.
6. In the **Set the selected certificate as active** dialog, confirm that you understand that setting a certificate to active may require you to re-establish the trust, and then choose **Make active**.

Step 4: Delete the old certificate.

Use the following procedure to complete the certificate rotation process for your application. You can only delete a certificate that is in an **Inactive** state.

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, choose your application.
4. On the application details page, select the **Configuration** tab. Under **IAM Identity Center metadata**, choose **Manage certificate**.
5. On the **IAM Identity Center certificate** page, select the certificate you want to delete. Choose **Actions** and then choose **Delete**.
6. In the **Delete certificate** dialog box, choose **Delete**.

Certificate expiration status indicators

While on the **Applications** page in the properties of an application, you may notice colored status indicator icons. These icons appear in the **Expires on** column next to each certificate in the list. The following describes the criteria that IAM Identity Center uses to determine which icon is displayed for each certificate.

- **Red** – Indicates that a certificate is currently expired.
- **Yellow** – Indicates that a certificate will expire in 90 days or less.
- **Green** – Indicates that a certificate is currently valid and will remain valid for at least 90 more days.

To check the current status of a certificate

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, review the status of the certificates in the list as indicated in the **Expires on** column.

Application properties in the IAM Identity Center console

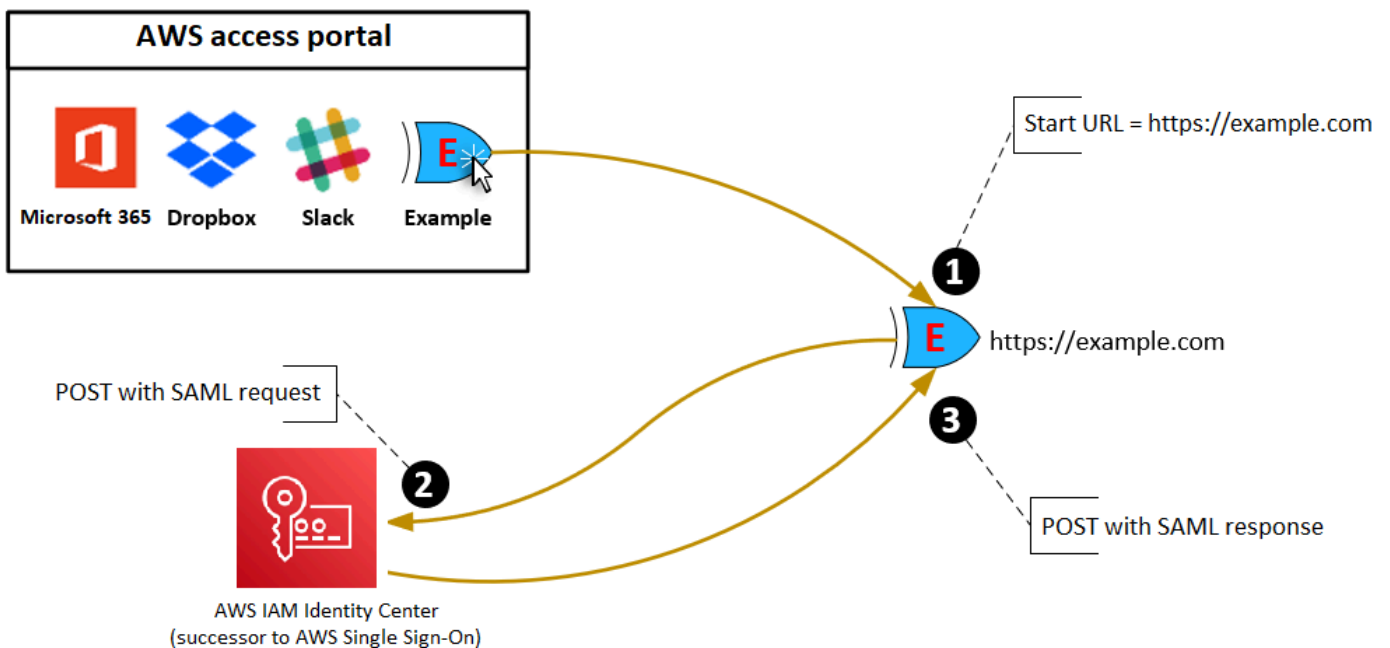
In IAM Identity Center you can customize the user experience by configuring the application start URL, relay state, and session duration.

Application start URL

You use an application start URL to start the federation process with your application. The typical use is for an application that supports only service provider (SP)-initiated binding.

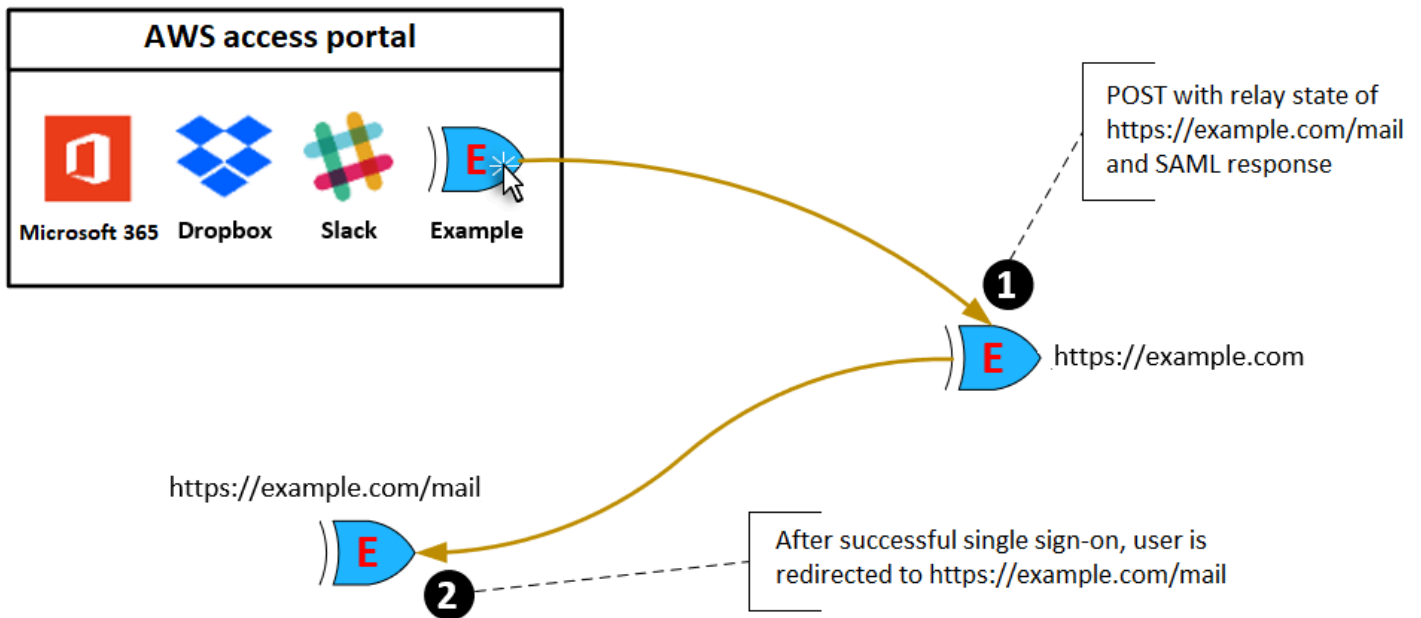
The following steps and diagram illustrate the application start URL authentication workflow when a user chooses an application in the AWS access portal:

1. The user's browser redirects the authentication request using the value for the application start URL (in this case `https://example.com`).
2. The application sends an HTML POST with a SAMLRequest to IAM Identity Center.
3. IAM Identity Center then sends an HTML POST with a SAMLResponse back to the application.



Relay state

During the federation authentication process, the relay state redirects users within the application. For SAML 2.0, this value is passed, unmodified, to the application. After the application properties are configured, IAM Identity Center sends the relay state value along with a SAML response to the application.



Session duration

Session duration is the length of time for which an application user session is valid. For SAML 2.0, this is used to set the `SessionNotOnOrAfter` date of the SAML assertion's element `saml2:AuthNStatement`.

Session duration can be interpreted by applications in either of the following ways:

- Applications can use it to determine the maximum time that is allowed for the user's session. Applications might generate a user session with a shorter duration. This can happen when the application only supports user sessions with a duration that is shorter than the configured session length.
- Applications can use it as the exact duration and might not allow administrators to configure the value. This can happen when the application only supports a specific session length.

For more information about how session duration is used, see your specific application's documentation.

Assign user access to applications in the IAM Identity Center console

You can assign users single sign-on access to SAML 2.0 applications in the application catalog or to custom SAML 2.0 applications.

Considerations for group assignments:

- **Assign access directly to groups.** To help simplify administration of access permissions, we recommend that you assign access directly to groups rather than to individual users. With groups you can grant or deny permissions to groups of users, instead of applying those permissions to each individual. If a user moves to a different organization, you simply move that user to a different group. The user then automatically receives the permissions that are needed for the new organization.
- **Nested groups aren't supported.** When assigning user access to applications, IAM Identity Center doesn't support users being added to nested groups. If a user is added to a nested group, they might receive a "You do not have any applications" message during sign-in. Assignments must be made against the immediate group for which the user is a member.

To assign user or group access to applications

Important

For AWS managed applications, you must add users directly from within the relevant application consoles or through the APIs.

1. Open the [IAM Identity Center console](#).

Note

If you manage users in AWS Managed Microsoft AD, make sure that the IAM Identity Center console is using the AWS Region where your AWS Managed Microsoft AD directory is located before taking the next step.

2. Choose **Applications**.
3. In the list of applications, choose the application name to which you want to assign access.

4. On the application details page, in the **Assigned users** section, choose **Assign users**.
5. In the **Assign users** dialog box, enter a user or group name. You can also search users and groups. You can specify multiple users or groups by selecting the applicable accounts as they appear in search results.
6. Choose **Assign users**.

Remove user access in the IAM Identity Center console

Use this procedure to remove user access to SAML 2.0 applications in the application catalog or custom SAML 2.0 applications.

To remove user access to an application

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, choose the application from which you want to remove user access.
4. On the application details page, in the **Assigned users** section, select the user or group that you want to remove and then choose the **Remove access** button.
5. In the **Remove access** dialog box, verify the user or group name. Then choose **Remove access**.

Map attributes in your application to IAM Identity Center attributes

Some service providers require custom SAML assertions to pass additional data about your user sign-ins. In that case, use the following procedure to specify how your applications user attributes should map to corresponding attributes in IAM Identity Center.

To map application attributes to attributes in IAM Identity Center

1. Open the [IAM Identity Center console](#).
2. Choose **Applications**.
3. In the list of applications, choose the application where you want to map attributes.
4. On the application details page, choose the **Actions** and then choose **Edit attribute mapping**.
5. Choose **Add new attribute mapping**.

6. In the first text box, enter the application attribute.
7. In the second text box, enter the attribute in IAM Identity Center that you want to map to the application attribute. For example, you might want to map the application attribute **Username** to the IAM Identity Center user attribute **email**. To see the list of allowed user attributes in IAM Identity Center, see the table in [Attribute mappings for AWS Managed Microsoft AD directory](#).
8. In the third column of the table, choose the appropriate format for the attribute from the menu.
9. Choose **Save changes**.

Resiliency design and Regional behavior

The IAM Identity Center service is fully managed and uses highly available and durable AWS services, such as Amazon S3 and Amazon EC2. To ensure availability in the event of an availability zone disruption, IAM Identity Center operates across multiple availability zones.

You enable IAM Identity Center in your AWS Organizations management account. This is required so that IAM Identity Center can provision, de-provision, and update roles across all your AWS accounts. When you enable IAM Identity Center, it's deployed to the AWS Region that's currently selected. If you want to deploy to a specific AWS Region, change the region selection before enabling IAM Identity Center.

Note

IAM Identity Center controls access to its permission sets and applications from its primary Region only. We recommend that you consider the risks associated with access control when IAM Identity Center operates in a single Region.

Although IAM Identity Center determines access from the Region in which you enable the service, AWS accounts are global. This means that after users sign in to IAM Identity Center, they can operate in any Region when they access AWS accounts through IAM Identity Center. Most AWS managed applications such as Amazon SageMaker, however, must be installed in the same Region as IAM Identity Center for users to authenticate and assign access to these applications. For information about Regional constraints when using an application with IAM Identity Center, see the documentation for the application.

You can also use IAM Identity Center to authenticate and authorize access to SAML-based applications that are reachable through a public URL, regardless of the platform or cloud on which the application is built.

We don't recommend using [Account instances of IAM Identity Center](#) as a means to implement resiliency as it creates a second, isolated control point that isn't connected to your organization instance.

Designed for availability

The following table provides the availability that IAM Identity Center is designed to achieve. These values don't represent a Service Level Agreement or guarantee, but rather provide insight to the design goals. The availability percentages reference access to data or functions, and aren't a reference to durability (for example, long term retention of data).

Service component	Availability design goal
Data plane (including sign-in)	99.95%
Control plane	99.90%

Set up emergency access to the AWS Management Console

IAM Identity Center is built from highly available AWS infrastructure and uses an Availability Zone architecture to eliminate single points of failure. For an extra layer of protection in the unlikely event of an IAM Identity Center or AWS Region disruption, we recommend that you set up a configuration that you can use to provide temporary access to the AWS Management Console.

Contents

- [Overview](#)
- [Summary of emergency access configuration](#)
- [How to design your critical operations roles](#)
- [How to plan your access model](#)
- [How to design emergency role, account, and group mapping](#)
- [How to create your emergency access configuration](#)
- [Emergency preparation tasks](#)
- [Emergency failover process](#)
- [Return to normal operations](#)
- [One-time setup of a direct IAM federation application in Okta](#)

Overview

AWS enables you to:

- [Connect your third-party IdP to IAM Identity Center.](#)
- Connect your third-party IdP to individual AWS accounts by using [SAML 2.0-based federation.](#)

If you use IAM Identity Center, you can use these capabilities to create the emergency access configuration described in the following sections. This configuration enables you to use IAM Identity Center as the mechanism for AWS account access. If IAM Identity Center is disrupted, your emergency operations users can sign in to the AWS Management Console through direct federation, by using the same credentials that they use to access their accounts. This configuration works when IAM Identity Center is unavailable, but the IAM data plane and your external identity provider (IdP) are available.

Important

We recommend that you deploy this configuration before a disruption occurs because you can't create the configuration if your access to create the required IAM roles is also disrupted. Also, test this configuration periodically to ensure that your team understands what to do if IAM Identity Center is disrupted.

Summary of emergency access configuration

To configure emergency access, you must complete the following tasks:

1. [Create an emergency operations account in your organization in AWS Organizations.](#)
2. Connect your IdP to the emergency operations account by using [SAML 2.0-based federation.](#)
3. In the emergency operations account, [create a role for third-party identity provider federation.](#) Also, create an emergency operations role in each of your workload accounts, with your required permissions.
4. [Delegate access to your workload accounts for the IAM role](#) that you created in the emergency operations account. To authorize access to your emergency operations account, create an emergency operations group in your IdP, with no members.

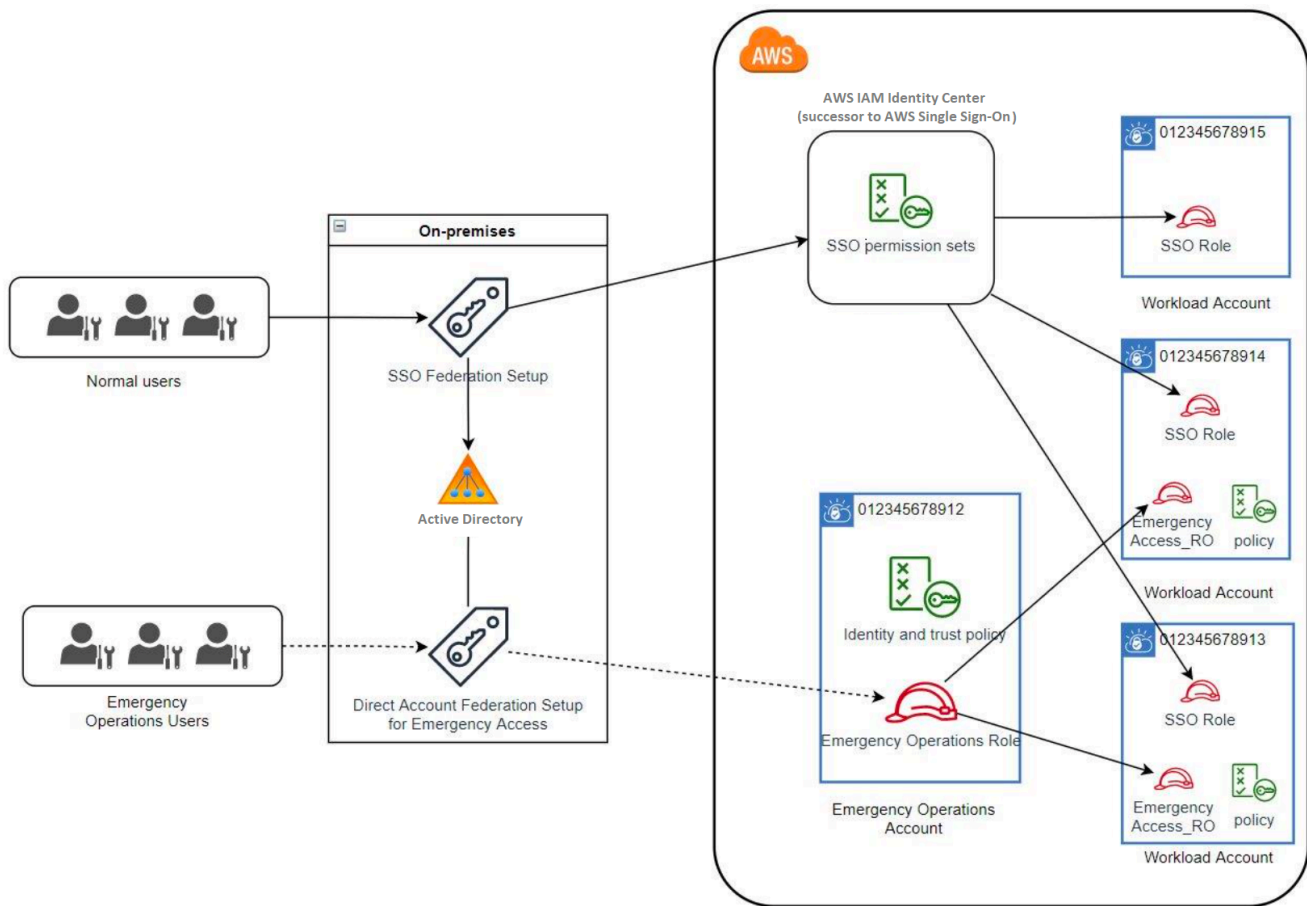
5. Enable the emergency operations group in your IdP to use the emergency operations role by creating a rule in your IdP that [enables SAML 2.0 federated access to the AWS Management Console](#).

During normal operations, no one has access to the emergency operations account because the emergency operations group in your IdP has no members. In the event of an IAM Identity Center disruption, use your IdP to add trusted users to the emergency operations group in your IdP. These users can then sign in to your IdP, navigate to the AWS Management Console, and assume the emergency operations role in the emergency operations account. From there, these users can [switch roles](#) to the emergency access role in your workload accounts where they need to perform operations work.

How to design your critical operations roles

With this design, you configure a single AWS account in which you federate through IAM, so that users can assume critical operations roles. The critical operations roles have a trust policy that enables users to assume a corresponding role in your workload accounts. The roles in the workload accounts provide the permissions that users require to perform essential work.

The following diagram provides a design overview.



How to plan your access model

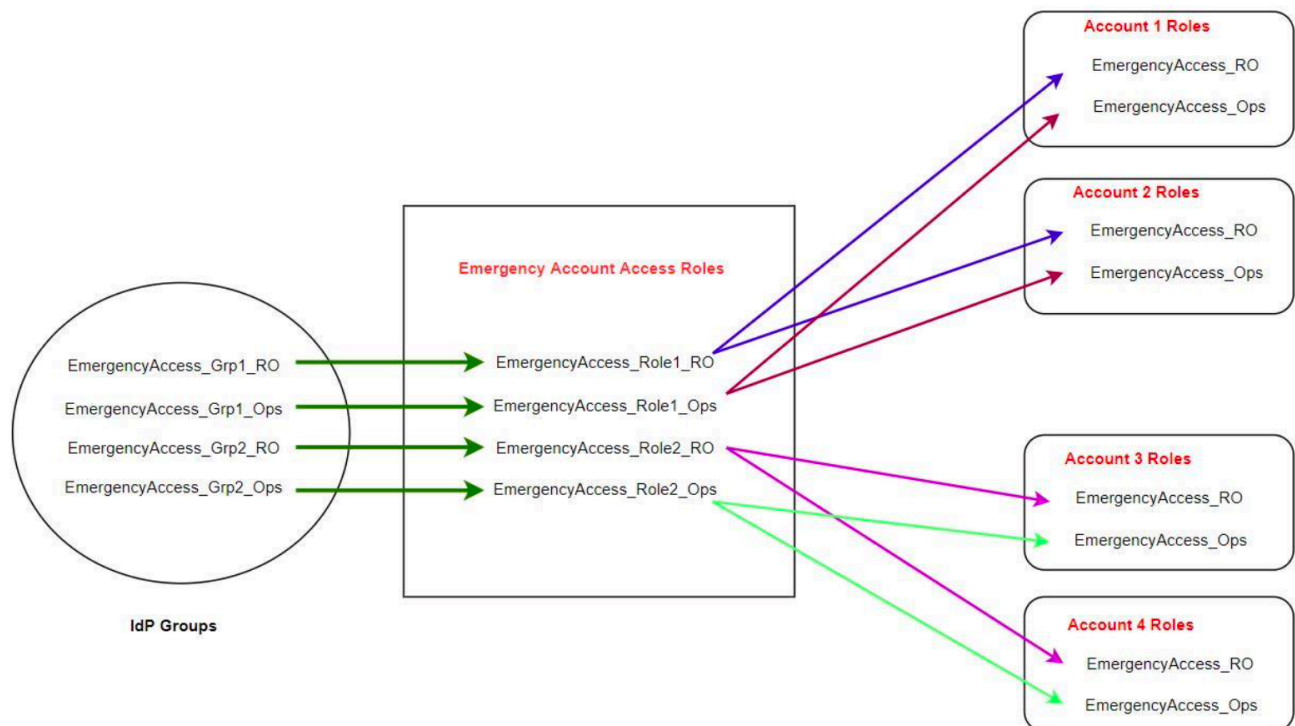
Before you configure emergency access, create a plan for how the access model will work. Use the following process to create this plan.

1. Identify the AWS accounts where emergency operator access is essential during a disruption to IAM Identity Center. For example, your production accounts are probably essential, but your development and test accounts might not be.
2. For that collection of accounts, identify the specific critical roles that you need in your accounts. Across these accounts, be consistent in defining what the roles can do. This simplifies work in your emergency access account where you create cross-account roles. We recommend that you start with two distinct roles in these accounts: Read Only (RO) and Operations (Ops). If required, you can create more roles and map these roles to a more distinct group of emergency access users in your setup.

3. Identify and create emergency access groups in your IdP. The group members are the users to whom you are delegating access to emergency access roles.
4. Define which roles these groups can assume in the emergency access account. To do this, define rules in your IdP that generate claims that list which roles the group can access. These groups can then assume your Read Only or Operations roles in emergency access account. From those roles, they can assume corresponding roles in your workload accounts.

How to design emergency role, account, and group mapping

The following diagram shows how to map your emergency access groups to roles in your emergency access account. The diagram also shows the cross-account role trust relationships that enable emergency access account roles to access corresponding roles in your workload accounts. We recommend that your emergency plan design use these mappings as a starting point.



How to create your emergency access configuration

Use the following mapping table to create your emergency access configuration. This table reflects a plan that includes two roles in the workload accounts: Read Only (RO) and Operations (Ops) , with corresponding trust policies and permissions policies. The trust policies enable the

emergency access account roles to access the individual workload account roles. The individual workload account roles also have permissions policies for what the role can do in the account. The permissions policies can be [AWS managed policies](#) or [customer managed policies](#).

Account	Roles to create	Trust policy	Permissions policy
Account 1	EmergencyAccess_RO	EmergencyAccess_Role1_RO	arn:aws:iam::aws:policy/ReadOnlyAccess
Account 1	EmergencyAccess_Ops	EmergencyAccess_Role1_Ops	arn:aws:iam::aws:policy/job-function/SystemAdministrator
Account 2	EmergencyAccess_RO	EmergencyAccess_Role2_RO	arn:aws:iam::aws:policy/ReadOnlyAccess
Account 2	EmergencyAccess_Ops	EmergencyAccess_Role2_Ops	arn:aws:iam::aws:policy/job-function/SystemAdministrator
Emergency access account	EmergencyAccess_Role1_RO EmergencyAccess_Role1_Ops EmergencyAccess_Role2_RO EmergencyAccess_Role2_Ops	IdP	AssumeRole for role resource in account

In this mapping plan, the emergency access account contains two read-only roles and two operations roles. These roles trust your IdP to authenticate and authorize your selected groups to access the roles by passing the names of the roles in assertions. There are corresponding read-only and operations roles in workload Account 1 and Account 2. For workload Account 1, the EmergencyAccess_RO role trusts the EmergencyAccess_Role1_RO role that resides in the

emergency access account. The table specifies similar trust patterns between the workload account read-only and operations roles and the corresponding emergency access roles.

Emergency preparation tasks

To prepare your emergency access configuration, we recommend that you perform the following tasks before an emergency occurs.

1. Set up a direct IAM federation application in your IdP. For more information, see [One-time setup of a direct IAM federation application in Okta](#).
2. Create an IdP connection in the emergency access account that can be accessed during the event.
3. Create emergency access roles in the emergency access accounts as described in the mapping table above.
4. Create temporary operations roles with trust and permission policies in each of the workload accounts.
5. Create temporary operations groups in your IdP. The group names will depend on the names of the temporary operations roles.
6. Test direct IAM federation.
7. Disable the IdP federation application in your IdP to prevent regular usage.

Emergency failover process

When an IAM Identity Center instance isn't available and you determine that you must provide emergency access to the AWS Management Console, we recommend the following failover process.

1. The IdP administrator enables the direct IAM federation application in your IdP.
2. Users request access to the temporary operations group through your existing mechanism, such as an email request, Slack channel, or other form of communication.
3. Users that you add to your emergency access groups sign in to the IdP, select the emergency access account, and, users choose a role to use in the emergency access account. From these roles, they can assume roles in corresponding workload accounts that have cross-account trust with the emergency account role.

Return to normal operations

Check the [AWS Health Dashboard](#) to confirm when the health of the IAM Identity Center service is restored. To return to normal operations, perform the following steps.

1. After the status icon for the IAM Identity Center service indicates that the service is healthy, sign in to IAM Identity Center.
2. If you can sign in to IAM Identity Center successfully, communicate to emergency access users that IAM Identity Center is available. Instruct these users to sign out and use the AWS access portal to sign back in to IAM Identity Center.
3. After all emergency access users sign out, in the IdP, disable the IdP federation application. We recommend that you perform this task after working hours.
4. Remove all users from the emergency access group in the IdP.

Your emergency access role infrastructure remains in place as a backup access plan, but it is now disabled.

One-time setup of a direct IAM federation application in Okta

1. Sign in to your Okta account as a user with administrative permissions.
2. In the Okta Admin Console, under **Applications**, choose **Applications**.
3. Choose **Browse App Catalog**. Search for and choose **AWS Account Federation**. Then choose **Add integration**.
4. Set up direct IAM federation with AWS by following the steps in [How to Configure SAML 2.0 for AWS Account Federation](#).
5. On the **Sign-On Options** tab, select SAML 2.0 and enter **Group Filter** and **Role Value Pattern** settings. The name of the group for the user directory depends on the filter that you configure.

Group Filter	<code>^aws\#\S+\#(?[role])[\w\-\-]+\#(?[accountid])\d+\$</code>
Role Value Pattern	<code>arn:aws:iam::[accountid]:saml-provider/Okta,arn:aws:iam::[accountid]:role/[role]</code>

In the figure above, the `role` variable is for the emergency operations role in your emergency access account. For example, if you create the `EmergencyAccess_Role1_R0` role (as

described in the mapping table) in AWS account 123456789012, and if your group filter setting is configured as shown in the figure above, your group name should be `aws#EmergencyAccess_Role1_R0#123456789012`.

6. In your directory (for example, your directory in Active Directory), create the emergency access group and specify a name for the directory (for example, `aws#EmergencyAccess_Role1_R0#123456789012`). Assign your users to this group by using your existing provisioning mechanism.
7. In the emergency access account, [configure a custom trust policy](#) that provides the permissions required for the emergency access role to be assumed during a disruption. Following is an example statement for a custom **trust policy** that is attached to the `EmergencyAccess_Role1_R0` role. For an illustration, see the emergency account in the diagram under [How to design emergency role, account, and group mapping](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::123456789012:saml-provider/Okta"
      },
      "Action": [
        "sts:AssumeRoleWithSAML",
        "sts:SetSourceIdentity",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "SAML:aud": "https://~/~/signin.aws.amazon.com/saml"
        }
      }
    }
  ]
}
```

8. The following is an example statement for a **permissions policy** that is attached to the `EmergencyAccess_Role1_R0` role. For an illustration, see the emergency account in the diagram under [How to design emergency role, account, and group mapping](#).

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": [
      "arn:aws:iam::<account 1>:role/EmergencyAccess_R0",
      "arn:aws:iam::<account 2>:role/EmergencyAccess_R0"
    ]
  }
]
}

```

9. On the workload accounts, configure a custom trust policy. Following is an example statement for a **trust policy** that is attached to the EmergencyAccess_R0 role. In this example, account 123456789012 is the emergency access account. For an illustration, see workload account in the diagram under [How to design emergency role, account, and group mapping](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Note

Most IdPs enable you to keep an application integration deactivated until required. We recommend that you keep the direct IAM federation application deactivated in your IdP until required for emergency access.

Security in AWS IAM Identity Center

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS compliance programs](#). To learn about the compliance programs that apply to AWS IAM Identity Center, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using IAM Identity Center. The following topics show you how to configure IAM Identity Center to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your IAM Identity Center resources.

Topics

- [Identity and access management for IAM Identity Center](#)
- [IAM Identity Center console and API authorization](#)
- [AWS STS condition context keys for IAM Identity Center](#)
- [Logging and monitoring in IAM Identity Center](#)
- [Compliance validation for IAM Identity Center](#)
- [Resilience in IAM Identity Center](#)
- [Infrastructure security in IAM Identity Center](#)

Identity and access management for IAM Identity Center

Access to IAM Identity Center requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an AWS managed application.

Authentication to the AWS access portal is controlled by the directory that you have connected to IAM Identity Center. However, authorization to the AWS accounts that are available to users from within the AWS access portal is determined by two factors:

1. Who has been assigned access to those AWS accounts in the IAM Identity Center console. For more information, see [Single sign-on access to AWS accounts](#).
2. What level of permissions have been granted to the users in the IAM Identity Center console to allow them the appropriate access to those AWS accounts. For more information, see [Create, manage, and delete permission sets](#).

The following sections explain how you as an administrator can control access to the IAM Identity Center console or can delegate administrative access for day-to-day tasks from the IAM Identity Center console.

- [Authentication](#)
- [Access control](#)

Authentication

Learn how to access AWS using [IAM identities](#).

Access control

You can have valid credentials to authenticate your requests, but unless you have permissions, you can't create or access IAM Identity Center resources. For example, you must have permissions to create an IAM Identity Center connected directory.

The following sections describe how to manage permissions for IAM Identity Center. We recommend that you read the overview first.

- [Overview of managing access permissions to your IAM Identity Center resources](#)

- [Identity-based policy examples for IAM Identity Center](#)
- [Using service-linked roles for IAM Identity Center](#)

Overview of managing access permissions to your IAM Identity Center resources

Every AWS resource is owned by an AWS account, and permissions to create or access the resources are governed by permissions policies. To provide access, an account administrator can add permissions to IAM identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support adding permissions to resources.

Note

An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see [IAM best practices](#) in the *IAM User Guide*.

Topics

- [IAM Identity Center resources and operations](#)
- [Understanding resource ownership](#)
- [Managing access to resources](#)
- [Specifying policy elements: actions, effects, resources, and principals](#)
- [Specifying conditions in a policy](#)

IAM Identity Center resources and operations

In IAM Identity Center, the primary resources are application instances, profiles, and permission sets.

Understanding resource ownership

A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the account, a user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If the AWS account root user creates an IAM Identity Center resource, such as an application instance or permission set, your AWS account is the owner of that resource.
- If you create a user in your AWS account and grant that user permissions to create IAM Identity Center resources, the user can then create IAM Identity Center resources. However, your AWS account, to which the user belongs, owns the resources.
- If you create an IAM role in your AWS account with permissions to create IAM Identity Center resources, anyone who can assume the role can create IAM Identity Center resources. Your AWS account, to which the role belongs, owns the IAM Identity Center resources.

Managing access to resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

Note

This section discusses using IAM in the context of IAM Identity Center. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see [What is IAM?](#) in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see [AWS IAM policy reference](#) in the *IAM User Guide*.

Policies that are attached to an IAM identity are referred to as *identity-based* policies (IAM policies). Policies that are attached to a resource are referred to as *resource-based* policies. IAM Identity Center supports only identity-based policies (IAM policies).

Topics

- [Identity-based policies \(IAM policies\)](#)
- [Resource-based policies](#)

Identity-based policies (IAM policies)

You can add permissions to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your AWS account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to add an IAM Identity Center resource, such as a new application.

- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions.

For more information about using IAM to delegate permissions, see [Access management](#) in the *IAM User Guide*.

The following permissions policy grants permissions to a user to run all of the actions that begin with `List`. These actions show information about an IAM Identity Center resource, such as an application instance or permissions set. Note that the wildcard character (*) in the `Resource` element indicates that the actions are allowed for all IAM Identity Center resources that are owned by the account.

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

For more information about using identity-based policies with IAM Identity Center, see [Identity-based policy examples for IAM Identity Center](#). For more information about users, groups, roles, and permissions, see [Identities \(users, groups, and roles\)](#) in the *IAM User Guide*.

Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. IAM Identity Center doesn't support resource-based policies.

Specifying policy elements: actions, effects, resources, and principals

For each IAM Identity Center resource (see [IAM Identity Center resources and operations](#)), the service defines a set of API operations. To grant permissions for these API operations, IAM Identity Center defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action.

The following are the basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies.
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `sso:DescribePermissionsPolicies` permission allows the user permissions to perform the IAM Identity Center `DescribePermissionsPolicies` operation.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). IAM Identity Center doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see [AWS IAM policy reference](#) in the *IAM User Guide*.

Specifying conditions in a policy

When you grant permissions, you can use the access policy language to specify the conditions that are required for a policy to take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see [Condition](#) in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to IAM Identity Center. However, there are AWS condition keys that you can use as appropriate. For a complete list of AWS keys, see [Available global condition keys](#) in the *IAM User Guide*.

Identity-based policy examples for IAM Identity Center

This topic provides examples of IAM policies that you can create to grant users and roles permissions to administer IAM Identity Center.

⚠ Important

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your IAM Identity Center resources. For more information, see [Overview of managing access permissions to your IAM Identity Center resources](#).

The sections in this topic cover the following:

- [Custom policy examples](#)
- [Permissions required to use the IAM Identity Center console](#)

Custom policy examples

This section provides examples of common use cases that require a custom IAM policy. These example policies are identity-based policies, which do not specify the Principal element. This is because with an identity-based policy, you don't specify the principal who gets the permission. Instead, you attach the policy to the principal. When you attach an identity-based permission policy to an IAM role, the principal identified in the role's trust policy gets the permissions. You can create identity-based policies in IAM and attach them to users, groups, and/or roles. You can also apply these policies to IAM Identity Center users when you create a permission set in IAM Identity Center.

ℹ Note

Use these examples when you create policies for your environment and make sure to test for both positive ("access granted") and negative ("access denied") test cases before you deploy these policies in your production environment. For more information about testing IAM policies, see [Testing IAM policies with the IAM policy simulator](#) in the *IAM User Guide*.

Topics

- [Example 1: Allow a user to view IAM Identity Center](#)
- [Example 2: Allow a user to manage permissions to AWS accounts in IAM Identity Center](#)
- [Example 3: Allow a user to manage applications in IAM Identity Center](#)
- [Example 4: Allow a user to manage users and groups in your Identity Center directory](#)

Example 1: Allow a user to view IAM Identity Center

The following permissions policy grants read-only permissions to a user so they can view all the settings and directory information configured in IAM Identity Center.

Note

This policy is provided for example purposes only. In a production environment, we recommend that you use the `ViewOnlyAccess` AWS managed policy for IAM Identity Center.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ds:DescribeDirectories",
        "ds:DescribeTrusts",
        "iam:ListPolicies",
        "organizations:DescribeOrganization",
        "organizations:DescribeAccount",
        "organizations:ListParents",
        "organizations:ListChildren",
        "organizations:ListAccounts",
        "organizations:ListRoots",
        "organizations:ListAccountsForParent",
        "organizations:ListOrganizationalUnitsForParent",
        "sso:ListManagedPoliciesInPermissionSet",
        "sso:ListPermissionSetsProvisionedToAccount",
        "sso:ListAccountAssignments",
        "sso:ListAccountsForProvisionedPermissionSet",
        "sso:ListPermissionSets",
        "sso:DescribePermissionSet",
        "sso:GetInlinePolicyForPermissionSet",
        "sso-directory:DescribeDirectory",
        "sso-directory:SearchUsers",
        "sso-directory:SearchGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

```

    }
  ]
}

```

Example 2: Allow a user to manage permissions to AWS accounts in IAM Identity Center

The following permissions policy grants permissions to allow a user to create, manage, and deploy permission sets for your AWS accounts.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:AttachManagedPolicyToPermissionSet",
        "sso:CreateAccountAssignment",
        "sso:CreatePermissionSet",
        "sso>DeleteAccountAssignment",
        "sso>DeleteInlinePolicyFromPermissionSet",
        "sso>DeletePermissionSet",
        "sso:DetachManagedPolicyFromPermissionSet",
        "sso:ProvisionPermissionSet",
        "sso:PutInlinePolicyToPermissionSet",
        "sso:UpdatePermissionSet"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IAMListPermissions",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:ListPolicies"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AccessToSSOProvisionedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",

```

```

        "iam:DeleteRole",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:ListRolePolicies",
        "iam:PutRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/*"
},
{
    "Effect": "Allow",
    "Action": [
        "iam:GetSAMLProvider"
    ],
    "Resource": "arn:aws:iam::*:saml-provider/AWSSSO*_DO_NOT_DELETE"
}
]
}

```

Note

The additional permissions listed under the "Sid": "IAMListPermissions", and "Sid": "AccessToSSOProvisionedRoles" sections are required only to enable the user to create assignments in the AWS Organizations management account. In certain cases, you may also need to add `iam:UpdateSAMLProvider` to these sections.

Example 3: Allow a user to manage applications in IAM Identity Center

The following permissions policy grants permissions to allow a user to view and configure applications in IAM Identity Center, including pre-integrated SaaS applications from within the IAM Identity Center catalog.

Note

The `sso:AssociateProfile` operation used in the following policy example is required for management of user and group assignments to applications. It also allows a user to assign users and groups to AWS accounts by using existing permission sets. If a user must

manage AWS account access within IAM Identity Center, and requires permissions necessary to manage permission sets, see [Example 2: Allow a user to manage permissions to AWS accounts in IAM Identity Center](#).

As of October 2020, many of these operations are available only through the AWS console. This example policy includes “read” actions such as list, get, and search, which are relevant to the error-free operation of the console for this case.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:AssociateProfile",
        "sso:CreateApplicationInstance",
        "sso:ImportApplicationInstanceServiceProviderMetadata",
        "sso>DeleteApplicationInstance",
        "sso>DeleteProfile",
        "sso:DisassociateProfile",
        "sso:GetApplicationTemplate",
        "sso:UpdateApplicationInstanceServiceProviderConfiguration",
        "sso:UpdateApplicationInstanceDisplayData",
        "sso>DeleteManagedApplicationInstance",
        "sso:UpdateApplicationInstanceStatus",
        "sso:GetManagedApplicationInstance",
        "sso:UpdateManagedApplicationInstanceStatus",
        "sso:CreateManagedApplicationInstance",
        "sso:UpdateApplicationInstanceSecurityConfiguration",
        "sso:UpdateApplicationInstanceResponseConfiguration",
        "sso:GetApplicationInstance",
        "sso:CreateApplicationInstanceCertificate",
        "sso:UpdateApplicationInstanceResponseSchemaConfiguration",
        "sso:UpdateApplicationInstanceActiveCertificate",
        "sso>DeleteApplicationInstanceCertificate",
        "sso:ListApplicationInstanceCertificates",
        "sso:ListApplicationTemplates",
        "sso:ListApplications",
        "sso:ListApplicationInstances",
        "sso:ListDirectoryAssociations",
        "sso:ListProfiles",
```

```

        "sso:ListProfileAssociations",
        "sso:ListInstances",
        "sso:GetProfile",
        "sso:GetSSOStatus",
        "sso:GetSsoConfiguration",
        "sso-directory:DescribeDirectory",
        "sso-directory:DescribeUsers",
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
    ],
    "Resource": "*"
}
]
}

```

Example 4: Allow a user to manage users and groups in your Identity Center directory

The following permissions policy grants permissions to allow a user to create, view, modify, and delete users and groups in IAM Identity Center.

In some cases, direct modifications to users and groups in IAM Identity Center are restricted. For example, when Active Directory, or an external identity provider with Automatic Provisioning enabled, is selected as the identity source.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso-directory:ListGroupsWithUser",
        "sso-directory:DisableUser",
        "sso-directory:EnableUser",
        "sso-directory:SearchGroups",
        "sso-directory>DeleteGroup",
        "sso-directory:AddMemberToGroup",
        "sso-directory:DescribeDirectory",
        "sso-directory:UpdateUser",
        "sso-directory:ListMembersInGroup",
        "sso-directory:CreateUser",
        "sso-directory:DescribeGroups",
        "sso-directory:SearchUsers",

```

```

        "sso:ListDirectoryAssociations",
        "sso-directory:RemoveMemberFromGroup",
        "sso-directory:DeleteUser",
        "sso-directory:DescribeUsers",
        "sso-directory:UpdateGroup",
        "sso-directory:CreateGroup"
    ],
    "Resource": "*"
}
]
}

```

Permissions required to use the IAM Identity Center console

For a user to work with the IAM Identity Center console without errors, additional permissions are required. If an IAM policy has been created that is more restrictive than the minimum required permissions, the console won't function as intended for users with that policy. The following example lists the set of permissions that might be needed to ensure error-free operation within the IAM Identity Center console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:DescribeAccountAssignmentCreationStatus",
        "sso:DescribeAccountAssignmentDeletionStatus",
        "sso:DescribePermissionSet",
        "sso:DescribePermissionSetProvisioningStatus",
        "sso:DescribePermissionsPolicies",
        "sso:DescribeRegisteredRegions",
        "sso:GetApplicationInstance",
        "sso:GetApplicationTemplate",
        "sso:GetInlinePolicyForPermissionSet",
        "sso:GetManagedApplicationInstance",
        "sso:GetMfaDeviceManagementForDirectory",
        "sso:GetPermissionSet",
        "sso:GetPermissionsPolicy",
        "sso:GetProfile",
        "sso:GetSharedSsoConfiguration",
        "sso:GetSsoConfiguration",
        "sso:GetSSOStatus",

```

```

        "sso:GetTrust",
        "sso:ListAccountAssignmentCreationStatus",
        "sso:ListAccountAssignmentDeletionStatus",
        "sso:ListAccountAssignments",
        "sso:ListAccountsForProvisionedPermissionSet",
        "sso:ListApplicationInstanceCertificates",
        "sso:ListApplicationInstances",
        "sso:ListApplications",
        "sso:ListApplicationTemplates",
        "sso:ListDirectoryAssociations",
        "sso:ListInstances",
        "sso:ListManagedPoliciesInPermissionSet",
        "sso:ListPermissionSetProvisioningStatus",
        "sso:ListPermissionSets",
        "sso:ListPermissionSetsProvisionedToAccount",
        "sso:ListProfileAssociations",
        "sso:ListProfiles",
        "sso:ListTagsForResource",
        "sso-directory:DescribeDirectory",
        "sso-directory:DescribeGroups",
        "sso-directory:DescribeUsers",
        "sso-directory:ListGroupsForUser",
        "sso-directory:ListMembersInGroup",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers"
    ],
    "Resource": "*"
}
]
}

```

AWS managed policies for IAM Identity Center

To [create IAM customer managed policies](#) that provide your team with only the permissions they need takes time and expertise. To get started quickly, you can use AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when

a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

New actions that allow you to list and delete user sessions are available under the new namespace `identitystore-auth`. Any additional permissions for actions in this namespace will be updated on this page. When creating your custom IAM policies, avoid using `*` after `identitystore-auth` because this applies to all actions that exist in the namespace today or in the future.

AWS managed policy: AWSSSOMasterAccountAdministrator

The `AWSSSOMasterAccountAdministrator` policy provides required administrative actions to principals. The policy is intended for principals who perform the job role of an AWS IAM Identity Center administrator. Over time the list of actions provided will be updated to match the existing functionality of IAM Identity Center and the actions that are required as an administrator.

You can attach the `AWSSSOMasterAccountAdministrator` policy to your IAM identities. When you attach the `AWSSSOMasterAccountAdministrator` policy to an identity, you grant administrative AWS IAM Identity Center permissions. Principals with this policy can access IAM Identity Center within the AWS Organizations management account and all member accounts. This principal can fully manage all IAM Identity Center operations, including the ability to create an IAM Identity Center instance, users, permission sets, and assignments. The principal can also instantiate those assignments throughout the AWS organization member accounts and establish connections between AWS Directory Service managed directories and IAM Identity Center. As new administrative features are released, the account administrator will be granted these permissions automatically.

Permissions groupings

This policy is grouped into statements based on the set of permissions provided.

- `AWSSSOMasterAccountAdministrator` – Allows IAM Identity Center to [pass the service role](#) named `AWSServiceRoleforSSO` to IAM Identity Center so that it can later assume the role and perform actions on their behalf. This is necessary when the person or application attempts to enable IAM Identity Center. For more information, see [Manage access to AWS accounts](#).

- `AWSSSOMemberAccountAdministrator` – Allows IAM Identity Center to perform account administrator actions in a multi-account AWS environment. For more information, see [AWS managed policy: AWSSSOMemberAccountAdministrator](#).
- `AWSSSOManageDelegatedAdministrator` – Allows IAM Identity Center to register and deregister a delegated administrator for your organization.

To view the permissions for this policy, see [AWSSSOMasterAccountAdministrator](#) in *AWS Managed Policy Reference*.

Additional information about this policy

When IAM Identity Center is enabled for the first time, the IAM Identity Center service creates a [service linked role](#) in the AWS Organizations management account (formerly master account) so that IAM Identity Center can manage the resources in your account. The actions required are `iam:CreateServiceLinkedRole` and `iam:PassRole`, which are shown in the following snippets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSSSOCreateSLR",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/sso.amazonaws.com/AWSServiceRoleForSSO",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "sso.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AWSSSOMasterAccountAdministrator",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/sso.amazonaws.com/AWSServiceRoleForSSO",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "sso.amazonaws.com"
        }
      }
    }
  ]
}
```

```
    }  
  },  
]  
}
```

AWS managed policy: AWSSSOMemberAccountAdministrator

The `AWSSSOMemberAccountAdministrator` policy provides required administrative actions to principals. The policy is intended for principals who perform the job role of an IAM Identity Center administrator. Over time the list of actions provided will be updated to match the existing functionality of IAM Identity Center and the actions that are required as an administrator.

You can attach the `AWSSSOMemberAccountAdministrator` policy to your IAM identities. When you attach the `AWSSSOMemberAccountAdministrator` policy to an identity, you grant administrative AWS IAM Identity Center permissions. Principals with this policy can access IAM Identity Center within the AWS Organizations management account and all member accounts. This principal can fully manage all IAM Identity Center operations, including the ability to create users, permission sets, and assignments. The principal can also instantiate those assignments throughout the AWS organization member accounts and establish connections between AWS Directory Service managed directories and IAM Identity Center. As new administrative features are released, the account administrator is granted these permissions automatically.

To view the permissions for this policy, see [AWSSSOMemberAccountAdministrator](#) in *AWS Managed Policy Reference*.

Additional information about this policy

IAM Identity Center administrators manage users, groups, and passwords in their Identity Center directory store (`sso-directory`). The account admin role includes permissions for the following actions:

- `"sso:*"`
- `"sso-directory:*"`

IAM Identity Center administrators need limited permissions to the following AWS Directory Service actions to perform daily tasks.

- `"ds:DescribeTrusts"`

- "ds:UnauthorizeApplication"
- "ds:DescribeDirectories"
- "ds:AuthorizeApplication"
- "ds:CreateAlias"

These permissions allow IAM Identity Center administrators to identify existing directories and manage applications so that they can be configured for use with IAM Identity Center. For more information about each of these actions, see [AWS Directory Service API permissions: Actions, resources, and conditions reference](#).

IAM Identity Center uses IAM policies to grant permissions to IAM Identity Center users. IAM Identity Center administrators create permission sets and attach policies to them. The IAM Identity Center administrator must have the permissions to list the existing policies so that they can choose which policies to use with the permission set they are creating or updating. To set secure and functional permissions, the IAM Identity Center administrator must have permissions to run the IAM Access Analyzer policy validation.

- "iam:ListPolicies"
- "access-analyzer:ValidatePolicy"

IAM Identity Center administrators need limited access to the following AWS Organizations actions to perform daily tasks:

- "organizations:EnableAWSServiceAccess"
- "organizations:ListRoots"
- "organizations:ListAccounts"
- "organizations:ListOrganizationalUnitsForParent"
- "organizations:ListAccountsForParent"
- "organizations:DescribeOrganization"
- "organizations:ListChildren"
- "organizations:DescribeAccount"
- "organizations:ListParents"
- "organizations:ListDelegatedAdministrators"
- "organizations:RegisterDelegatedAdministrator"

- "organizations:DeregisterDelegatedAdministrator"

These permissions allow IAM Identity Center administrators the ability to work with organization resources (accounts) for basic IAM Identity Center administrative tasks such as the following:

- Identifying the management account that belongs to the organization
- Identifying the member accounts that belong to the organization
- Enabling AWS service access for accounts
- Setting up and managing a delegated administrator

For more information about using a delegated administrator with IAM Identity Center, see [Delegated administration](#). For more information about how these permissions are used with AWS Organizations, see [Using AWS Organizations with other AWS services](#).

AWS managed policy: AWSSSODirectoryAdministrator

You can attach the `AWSSSODirectoryAdministrator` policy to your IAM identities.

This policy grants administrative permissions over IAM Identity Center users and groups. Principals with this policy attached can make any updates to IAM Identity Center users and groups.

To view the permissions for this policy, see [AWSSSODirectoryAdministrator](#) in *AWS Managed Policy Reference*.

AWS managed policy: AWSSSOReadOnly

You can attach the `AWSSSOReadOnly` policy to your IAM identities.

This policy grants read-only permissions that allow users to view information in IAM Identity Center. Principals with this policy attached cannot view the IAM Identity Center users or groups directly. Principals with this policy attached cannot make any updates in IAM Identity Center. For example, principals with these permissions can view IAM Identity Center settings, but cannot change any of the setting values.

To view the permissions for this policy, see [AWSSSOReadOnly](#) in *AWS Managed Policy Reference*.

AWS managed policy: AWSSSODirectoryReadOnly

You can attach the `AWSSSODirectoryReadOnly` policy to your IAM identities.

This policy grants read-only permissions that allow users to view users and groups in IAM Identity Center. Principals with this policy attached cannot view IAM Identity Center assignments, permission sets, applications, or settings. Principals with this policy attached can't make any updates in IAM Identity Center. For example, principals with these permissions can view IAM Identity Center users, but they can't change any user attributes or assign MFA devices.

To view the permissions for this policy, see [AWSSSODirectoryReadOnly](#) in *AWS Managed Policy Reference*.

AWS managed policy: AWSIdentitySyncFullAccess

You can attach the `AWSIdentitySyncFullAccess` policy to your IAM identities.

Principals with this policy attached have full access permissions to create and delete sync profiles, associate or update a sync profile with a sync target, create, list and delete sync filters, and start or stop synchronization.

Permission details

To view the permissions for this policy, see [AWSIdentitySyncFullAccess](#) in *AWS Managed Policy Reference*.

AWS managed policy: AWSIdentitySyncReadOnlyAccess

You can attach the `AWSIdentitySyncReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions that allow users to view information about the identity synchronization profile, filters, and target settings. Principals with this policy attached can't make any updates to synchronization settings. For example, principals with these permissions can view identity synchronization settings, but can't change any of the profile or filter values.

To view the permissions for this policy, see [AWSIdentitySyncReadOnlyAccess](#) in *AWS Managed Policy Reference*.

AWS managed policy: AWSSSOServiceRolePolicy

You can't attach the `AWSSSOServiceRolePolicy` policy to your IAM identities.

This policy is attached to a service-linked role that allows IAM Identity Center to delegate and enforce which users have single sign-on access to specific AWS accounts in AWS Organizations. When you enable IAM, a service-linked role is created in all of the AWS accounts within your organization. IAM Identity Center also creates the same service-linked role in every account that

is subsequently added to your organization. This role allows IAM Identity Center to access each account's resources on your behalf. Service-linked roles that are created in each AWS account are named `AWSServiceRoleForSSO`. For more information, see [Using service-linked roles for IAM Identity Center](#).

AWS managed policy: `AWSIAMIdentityCenterAllowListForIdentityContext`

When assuming a role with the IAM Identity Center identity context, AWS Security Token Service (AWS STS) automatically attaches the `AWSIAMIdentityCenterAllowListForIdentityContext` policy to the role.

This policy provides the list of actions that are allowed when you use trusted identity propagation with roles that are assumed with the IAM Identity Center identity context. All other actions that are called with this context are blocked. The identity context is passed as `ProvidedContext`.

To view the permissions for this policy, see [AWSIAMIdentityCenterAllowListForIdentityContext](#) in *AWS Managed Policy Reference*.

IAM Identity Center updates to AWS managed policies

The following table describes the updates to AWS managed policies for IAM Identity Center since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the IAM Identity Center Document history page.

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the <code>aoss:APIAccessAll</code> , <code>es:ESHttpHead</code> , <code>es:ESHttpPost</code> , <code>es:ESHttpGet</code> , <code>es:ESHttpPatch</code> , <code>es:ESHttpDelete</code> , and <code>es:ESHttpPut</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.	July 12, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy now includes the <code>qapps:PredictQApp</code> , <code>qapps:ImportDocument</code> , <code>qapps:AssociateLibraryItemReview</code> , <code>qapps:DisassociateLibraryItemReview</code> , <code>qapps:GetQAppSession</code> , <code>qapps:UpdateQAppSession</code> , <code>qapps:GetQAppSessionMetadata</code> , <code>qapps:UpdateQAppSessionMetadata</code> , and <code>qapps:TagResource</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.</p>	<p>June 27, 2024</p>
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy now includes the <code>elasticmapreduce:AddJobFlowSteps</code> , <code>elasticmapreduce:DescribeCluster</code> , <code>elasticmapreduce:CancelSteps</code> , <code>elasticmapreduce:DescribeStep</code> , and <code>elasticmapreduce:ListSteps</code> actions to support trusted identity propagation in Amazon EMR.</p>	<p>May 17, 2024</p>

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy now includes the <code>qapps:CreateQApp</code> , <code>qapps:PredictProblemStatementFromConversation</code> , <code>qapps:PredictQAppFromProblemStatement</code> , <code>qapps:CopyQApp</code> , <code>qapps:GetQApp</code> , <code>qapps:ListQApps</code> , <code>qapps:UpdateQApp</code> , <code>qapps>DeleteQApp</code> , <code>qapps:AssociateQAppWithUser</code> , <code>qapps:DisassociateQAppFromUser</code> , <code>qapps:ImportDocumentToQApp</code> , <code>qapps:ImportDocumentToQAppSession</code> , <code>qapps:CreateLibraryItem</code> , <code>qapps:GetLibraryItem</code> , <code>qapps:UpdateLibraryItem</code> , <code>qapps:CreateLibraryItemReview</code> , <code>qapps:ListLibraryItems</code> , <code>qapps:CreateSubscriptionToken</code> , <code>qapps:StartQAppSession</code> , and <code>qapps:StopQAppSession</code> actions to support identity-aware console sessions for AWS</p>	<p>April 30, 2024</p>

Change	Description	Date
	managed applications that support these sessions.	
AWSSSOMasterAccountAdministrator	This policy now includes the <code>signin:CreateTrustedIdentityPropagationApplicationForConsole</code> and <code>signin:ListTrustedIdentityPropagationApplicationsForConsole</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.	April 26, 2024
AWSSSOMemberAccountAdministrator	This policy now includes the <code>signin:CreateTrustedIdentityPropagationApplicationForConsole</code> and <code>signin:ListTrustedIdentityPropagationApplicationsForConsole</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.	April 26, 2024

Change	Description	Date
AWSSSOReadOnly	This policy now includes the <code>signin:ListTrustedIdentityPropagationApplicationsForConsole</code> action to support identity-aware console sessions for AWS managed applications that support these sessions.	April 26, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the <code>qbusiness:PutFeedback</code> action to support identity-aware console sessions for AWS managed applications that support these sessions.	April 26, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the <code>q:StartConversation</code> , <code>q:SendMessage</code> , <code>q:ListConversations</code> , <code>q:GetConversation</code> , <code>q:StartTroubleshootingAnalysis</code> , <code>q:GetTroubleshootingResults</code> , <code>q:StartTroubleshootingResolutionExplanation</code> , and <code>q:UpdateTroubleshootingCommandResult</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.	April 24, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the <code>sts:SetContext</code> action to support identity-aware console sessions for AWS managed applications that support these sessions.	April 19, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy now includes the <code>qbusiness:Chat</code> , <code>qbusiness:ChatSync</code> , <code>qbusiness:ListConversations</code> , <code>qbusiness:ListMessages</code> , and <code>qbusiness>DeleteConversation</code> actions to support identity-aware console sessions for AWS managed applications that support these sessions.</p>	<p>April 11, 2024</p>
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy now includes the <code>s3:GetAccessGrantsInstanceForPrefix</code> and <code>s3:GetDataAccess</code> actions.</p>	<p>November 26, 2023</p>
AWSIAMIdentityCenterAllowListForIdentityContext	<p>This policy provides the list of actions that are allowed when you use trusted identity propagation with roles that are assumed with the IAM Identity Center identity context.</p>	<p>November 15, 2023</p>
AWSSSODirectoryReadOnly	<p>This policy now includes the new namespace <code>identitystore-auth</code> with new permissions to allow users to list and get sessions.</p>	<p>February 21, 2023</p>

Change	Description	Date
AWSSSOServiceRolePolicy	This policy now allows the UpdateSAMLProvider _ action to be taken on the management account.	October 20, 2022
AWSSSOMasterAccountAdministrator	This policy now includes the new namespace identitystore-auth with new permissions to allow the admin to list and delete sessions for a user.	October 20, 2022
AWSSSOMemberAccountAdministrator	This policy now includes the new namespace identitystore-auth with new permissions to allow the admin to list and delete sessions for a user.	October 20, 2022
AWSSSODirectoryAdministrator	This policy now includes the new namespace identitystore-auth with new permissions to allow the admin to list and delete sessions for a user.	October 20, 2022

Change	Description	Date
AWSSSOMasterAccountAdministrator	This policy now includes new permissions to call ListDelegatedAdministrators in AWS Organizations. This policy also now includes a subset of permissions <code>AWSSSOManageDelegatedAdministrator</code> that includes permissions to call RegisterDelegatedAdministrator and DeregisterDelegatedAdministrator .	August 16, 2022
AWSSSOMemberAccountAdministrator	This policy now includes new permissions to call ListDelegatedAdministrators in AWS Organizations. This policy also now includes a subset of permissions <code>AWSSSOManageDelegatedAdministrator</code> that includes permissions to call RegisterDelegatedAdministrator and DeregisterDelegatedAdministrator .	August 16, 2022

Change	Description	Date
AWSSSOReadOnly	This policy now includes new permissions to call ListDelegatedAdministrators in AWS Organizations.	August 11, 2022
AWSSSOServiceRolePolicy	This policy now includes new permissions to call DeleteRolePermissionsBoundary and PutRolePermissionsBoundary .	July 14, 2022
AWSSSOServiceRolePolicy	This policy now includes new permissions that allow calls to ListAWSServiceAccessForOrganization and ListDelegatedAdministrators in AWS Organizations.	May 11, 2022
AWSSSOMasterAccountAdministrator AWSSSOMemberAccountAdministrator AWSSSOReadOnly	Add IAM Access Analyzer permissions that allow a principal to use the policy checks for validation.	April 28, 2022

Change	Description	Date
AWSSSOMasterAccountAdministrator	<p>This policy now allows all IAM Identity Center Identity Store service actions.</p> <p>For information about the actions available in the IAM Identity Center Identity Store service, see the IAM Identity Center Identity Store API Reference.</p>	March 29, 2022
AWSSSOMemberAccountAdministrator	This policy now allows all IAM Identity Center Identity Store service actions.	March 29, 2022
AWSSSODirectoryAdministrator	This policy now allows all IAM Identity Center Identity Store service actions.	March 29, 2022
AWSSSODirectoryReadOnly	This policy now grants access to the IAM Identity Center Identity Store service read actions. This access is required to retrieve user and group information from the IAM Identity Center Identity Store service.	March 29, 2022
AWSIdentitySyncFullAccess	This policy allows full access to identity-sync permissions.	March 3, 2022
AWSIdentitySyncReadOnlyAccess	This policy grants read-only permissions that allow a principal to view identity-sync settings.	March 3, 2022

Change	Description	Date
AWSSSOReadOnly	This policy grants read-only permissions that allow a principal to view IAM Identity Center configuration settings.	August 4, 2021
IAM Identity Center started tracking changes	IAM Identity Center started tracking changes for AWS managed policies.	August 4, 2021

Using service-linked roles for IAM Identity Center

AWS IAM Identity Center uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to IAM Identity Center. It is predefined by IAM Identity Center and includes all the permissions that the service requires to call other AWS services on your behalf. For more information, see [Service-linked roles](#).

A service-linked role makes setting up IAM Identity Center easier because you don't have to manually add the necessary permissions. IAM Identity Center defines the permissions of its service-linked role, and unless defined otherwise, only IAM Identity Center can assume its role. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

For information about other services that support service-linked roles, see [AWS Services That Work with IAM](#) and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for IAM Identity Center

IAM Identity Center uses the service-linked role named **AWSServiceRoleForSSO** to grant IAM Identity Center permissions to manage AWS resources, including IAM roles, policies, and SAML IdP on your behalf.

The AWSServiceRoleForSSO service-linked role trusts the following services to assume the role:

- IAM Identity Center (service prefix: sso)

The `AWSServiceRoleForSSO` service-linked role permissions policy allows IAM Identity Center to complete the following on roles on the path `"/aws-reserved/sso.amazonaws.com/"` and with the name prefix `"AWSReservedSSO_"`:

- `iam:AttachRolePolicy`
- `iam:CreateRole`
- `iam>DeleteRole`
- `iam>DeleteRolePermissionsBoundary`
- `iam>DeleteRolePolicy`
- `iam:DetachRolePolicy`
- `iam:GetRole`
- `iam>ListRolePolicies`
- `iam:PutRolePolicy`
- `iam:PutRolePermissionsBoundary`
- `iam>ListAttachedRolePolicies`

The `AWSServiceRoleForSSO` service-linked role permissions policy allows IAM Identity Center to complete the following on SAML providers with name prefix as `"AWSSSO_"`:

- `iam:CreateSAMLProvider`
- `iam:GetSAMLProvider`
- `iam:UpdateSAMLProvider`
- `iam>DeleteSAMLProvider`

The `AWSServiceRoleForSSO` service-linked role permissions policy allows IAM Identity Center to complete the following on all organizations:

- `organizations:DescribeAccount`
- `organizations:DescribeOrganization`
- `organizations:ListAccounts`
- `organizations:ListAWSServiceAccessForOrganization`
- `organizations:ListDelegatedAdministrators`

The `AWSServiceRoleForSSO` service-linked role permissions policy allows IAM Identity Center to complete the following on all IAM roles (*):

- `iam:listRoles`

The `AWSServiceRoleForSSO` service-linked role permissions policy allows IAM Identity Center to complete the following on `arn:aws:iam::*:role/aws-service-role/sso.amazonaws.com/AWSServiceRoleForSSO`:

- `iam:GetServiceLinkedRoleDeletionStatus`
- `iam>DeleteServiceLinkedRole`

The role permissions policy allows IAM Identity Center to complete the following actions on resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMRoleProvisioningActions",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam>DeleteRolePermissionsBoundary",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription",
        "iam:UpdateAssumeRolePolicy"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgMasterAccountId": "${aws:PrincipalAccount}"
        }
      }
    }
  ],
}
```

```

    "Sid": "IAMRoleReadActions",
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:ListRoles"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "IAMRoleCleanupActions",
    "Effect": "Allow",
    "Action": [
        "iam:DeleteRole",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:ListRolePolicies",
        "iam:ListAttachedRolePolicies"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/*"
    ]
},
{
    "Sid": "IAMSLRCleanupActions",
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus",
        "iam:DeleteRole",
        "iam:GetRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-service-role/sso.amazonaws.com/
AWSServiceRoleForSSO"
    ]
},
{
    "Sid": "IAMsamlProviderCreationAction",
    "Effect": "Allow",
    "Action": [
        "iam:CreateSAMLProvider"
    ]
},

```

```

"Resource": [
  "arn:aws:iam::*:saml-provider/AWSSSO_*"
],
"Condition": {
  "StringNotEquals": {
    "aws:PrincipalOrgMasterAccountId": "${aws:PrincipalAccount}"
  }
},
{
  "Sid": "IAMSAMLProviderUpdateAction",
  "Effect": "Allow",
  "Action": [
    "iam:UpdateSAMLProvider"
  ],
  "Resource": [
    "arn:aws:iam::*:saml-provider/AWSSSO_*"
  ]
},
{
  "Sid": "IAMSAMLProviderCleanupActions",
  "Effect": "Allow",
  "Action": [
    "iam:DeleteSAMLProvider",
    "iam:GetSAMLProvider"
  ],
  "Resource": [
    "arn:aws:iam::*:saml-provider/AWSSSO_*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "organizations:DescribeAccount",
    "organizations:DescribeOrganization",
    "organizations:ListAccounts",
    "organizations:ListAWSServiceAccessForOrganization",
    "organizations:ListDelegatedAdministrators"
  ],
  "Resource": [
    "*"
  ]
},
{

```

```

    "Sid": "AllowUnauthAppForDirectory",
    "Effect": "Allow",
    "Action": [
        "ds:UnauthorizeApplication"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "AllowDescribeForDirectory",
    "Effect": "Allow",
    "Action": [
        "ds:DescribeDirectories",
        "ds:DescribeTrusts"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "AllowDescribeAndListOperationsOnIdentitySource",
    "Effect": "Allow",
    "Action": [
        "identitystore:DescribeUser",
        "identitystore:DescribeGroup",
        "identitystore:ListGroups",
        "identitystore:ListUsers"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

Creating a service-linked role for IAM Identity Center

You don't need to manually create a service-linked role. Once enabled, IAM Identity Center creates a service-linked role in all accounts within the organization in AWS Organizations. IAM Identity Center also creates the same service-linked role in every account that is subsequently added to your organization. This role allows IAM Identity Center to access each account's resources on your behalf.

Notes

- If you're signed in to the AWS Organizations management account, it uses your currently signed-in role and not the service-linked role. This prevents the escalation of privileges.
- When IAM Identity Center performs any IAM operations in the AWS Organizations management account, all operations happen using the credentials of the IAM principal. This enables the logs in CloudTrail to provide visibility of who made all privilege changes in the management account.

Important

If you were using the IAM Identity Center service before December 7, 2017, when it began supporting service-linked roles, then IAM Identity Center created the `AWSServiceRoleForSSO` role in your account. To learn more, see [A New Role Appeared in My IAM Account](#).

If you delete this service-link role and then need to create it again, you can use the same process to recreate the role in your account.

Editing a service-linked role for IAM Identity Center

IAM Identity Center does not allow you to edit the `AWSServiceRoleForSSO` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for IAM Identity Center

You don't need to manually delete the `AWSServiceRoleForSSO` role. When an AWS account is removed from an AWS organization, IAM Identity Center automatically cleans up the resources and deletes the service-linked role from that AWS account.

You can also use the IAM console, the IAM CLI, or the IAM API to manually delete the service-linked role. To do this, you must first manually clean up the resources for your service-linked role and then you can manually delete it.

Note

If the IAM Identity Center service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete IAM Identity Center resources used by the `AWSServiceRoleForSSO`

1. [Remove user and group access to an AWS account](#) for all users and groups that have access to the AWS account.
2. [Delete permission sets](#) that you have associated with the AWS account.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `AWSServiceRoleForSSO` service-linked role. For more information, see [Deleting a Service-Linked Role](#) in the *IAM User Guide*.

IAM Identity Center console and API authorization

Existing IAM Identity Center console APIs support dual authorization, which allows you to maintain use of existing API operations when newer APIs are available. If you have existing instances of IAM Identity Center that were created prior to November 15, 2023 and October 15th, 2020, you can use the following tables to determine which API operations now map to newer API operations that were released after those dates.

Topics

- [API actions after November 2023](#)
- [API actions after October 2020](#)

API actions after November 2023

Instances of IAM Identity Center that were created before November 15, 2023 honor both old and new API actions as long as there is no explicit deny on any of the actions. Instances created after November 15, 2023 use [newer API actions](#) for authorization in the IAM Identity Center console.

Console operation name used before November 15, 2023	API action used after November 15, 2023
AssociateProfile	CreateApplicationAssignment
CreateManagedApplicationInstance CreateApplicationInstance	CreateApplication
CreateManagedApplicationInstance	PutApplicationAuthenticationMethod
DeleteApplicationInstance DeleteManagedApplicationInstance	DeleteApplication
DeleteSSO	DeleteInstance
DisassociateProfile	DeleteApplicationAssignment
GetApplicationTemplate	DescribeApplicationProvider
GetManagedApplicationInstance	DescribeApplication
GetSharedSsoConfiguration	DescribeInstance
ListApplicationInstances	ListApplications
ListApplicationTemplates	ListApplicationProviders
ListDirectoryAssociations	DescribeInstance
ListProfileAssociations	ListApplicationAssignments

Console operation name used before November 15, 2023	API action used after November 15, 2023
UpdateApplicationInstanceDisplayData UpdateApplicationInstanceStatus UpdateManagedApplicationInstanceStatus	UpdateApplication

API actions after October 2020

Instances of IAM Identity Center that were created before October 15, 2020 honor both old and new API actions as long as there is no explicit deny on any of the actions. Instances created after October 15, 2020 use [newer API actions](#) for authorization in the IAM Identity Center console.

Operation name	API actions used before October 15, 2020	API actions used after October 15, 2020
AssociateProfile	AssociateProfile	CreateAccountAssignment
AttachManagedPolicy	PutPermissionsPolicy	AttachManagedPolicyToPermissionSet
CreatePermissionSet	CreatePermissionSet	CreatePermissionSet
DeleteApplicationInstanceForAWSAccount	DeleteApplicationInstance DeleteTrust	DeleteAccountAssignment
DeleteApplicationProfileForAwsAccount	DeleteProfile	DeleteAccountAssignment
DeletePermissionsPolicy	DeletePermissionsPolicy	DeleteInlinePolicyFromPermissionSet
DeletePermissionSet	DeletePermissionSet	DeletePermissionSet
DescribePermissionsPolicies	DescribePermissionsPolicies	ListManagedPoliciesInPermissionSet

Operation name	API actions used before October 15, 2020	API actions used after October 15, 2020
DetachManagedPolicy	DeletePermissionsPolicy	DetachManagedPolicyFromPermissionSet
DisassociateProfile	DisassociateProfile	DeleteAccountAssignment
GetApplicationInstanceForAWSAccount	GetApplicationInstance	ListAccountAssignments
GetAWSAccountProfileStatus	GetProfile	ListPermissionSetsProvisionedToAccount
GetPermissionSet	GetPermissionSet	DescribePermissionSet
GetPermissionsPolicy	GetPermissionsPolicy	GetInlinePolicyForPermissionSet
ListAccountsWithProvisionedPermissionSet	ListApplicationInstances GetApplicationInstance	ListAccountsForProvisionedPermissionSet
ListAWSAccountProfiles	ListProfiles GetProfile	ListPermissionSetsProvisionedToAccount
ListPermissionSets	ListPermissionSets	ListPermissionSets
ListProfileAssociations	ListProfileAssociations	ListAccountAssignments
ProvisionApplicationInstanceForAWSAccount	GetApplicationInstance CreateApplicationInstance	CreateAccountAssignment
ProvisionApplicationProfileForAWSAccountInstance	GetProfile CreateProfile UpdateProfile	CreateAccountAssignment
ProvisionSAMLProvider	GetTrust CreateTrust UpdateTrust	CreateAccountAssignment
PutPermissionsPolicy	PutPermissionsPolicy	PutInlinePolicyToPermissionSet

Operation name	API actions used before October 15, 2020	API actions used after October 15, 2020
UpdatePermissionSet	UpdatePermissionSet	UpdatePermissionSet

AWS STS condition context keys for IAM Identity Center

When a [principal](#) makes a [request](#) to AWS, AWS gathers the request information into a *request context*, which is used to evaluate and authorize the request. You can use the `Condition` element of a JSON policy to compare keys in the request context with key values that you specify in your policy. Request information is provided by different sources, including the principal making the request, the resource, the request it is made against, and the metadata about the request itself. *Service-specific condition keys* are defined for use with an individual AWS service.

IAM Identity Center includes an AWS STS context provider that enables AWS managed applications and third-party applications to add values for condition keys that are defined by IAM Identity Center. These keys are included in [IAM roles](#). The key values are set when an application passes a token to AWS STS. The application obtains the token that it passes to AWS STS in either of the following ways:

- During authentication with IAM Identity Center.
- After token exchange with a [trusted token issuer](#) for trusted identity propagation. In this case, the application obtains a token from a trusted token issuer and exchanges that token for a token from IAM Identity Center.

These keys are typically used by applications that integrate with trusted identity propagation. In some cases, when key values are present, you can use these keys in IAM policies that you create to allow or deny permissions.

For example, you might want to provide conditional access to a resource based on the value of the `UserId`. This value indicates which IAM Identity Center user is using the role. The example is similar to using `SourceId`. Unlike `SourceId`, however, the value for `UserId` represents a specific, verified user from the identity store. This value is present in the token that the application obtains and then passes to AWS STS. It is not a general purpose string that can contain arbitrary values.

Topics

- [identitystore:UserId](#)

- [identitystore:IdentityStoreArn](#)
- [identitycenter:ApplicationArn](#)
- [identitycenter:CredentialId](#)
- [identitycenter:InstanceArn](#)

identitystore:UserId

This context key is the `UserId` of the IAM Identity Center user who is the subject of the context assertion issued by IAM Identity Center. The context assertion is passed to AWS STS. You can use this key to compare the `UserId` of the IAM Identity Center user on behalf of whom the request is made with the identifier for the user that you specify in the policy.

- **Availability** – This key is included in the request context after a context assertion issued by IAM Identity Center is set, when a role is assumed using any AWS STS `assume-role` command in the AWS CLI or AWS STS `AssumeRole` API operation.
- **Data type** – [String](#)
- **Value type** – Single-valued

identitystore:IdentityStoreArn

This context key is the ARN of the identity store that is attached to the instance of IAM Identity Center that issued the context assertion. It is also the identity store in which you can look up attributes for `identitystore:UserID`. You can use this key in policies to determine whether the `identitystore:UserID` comes from an expected identity store ARN.

- **Availability** – This key is included in the request context after a context assertion issued by IAM Identity Center is set, when a role is assumed using any AWS STS `assume-role` command in the AWS CLI or AWS STS `AssumeRole` API operation.
- **Data type** – [Arn](#), [String](#)
- **Value type** – Single-valued

identitycenter:ApplicationArn

This context key is the ARN of the application to which IAM Identity Center issued a context assertion. You can use this key in policies to determine whether

`identitycenter:ApplicationArn` comes from an expected application. Using this key can help prevent an IAM role from being accessed by an unexpected application.

- **Availability** – This key is included in the request context of an AWS STS `AssumeRole` API operation. The request context includes a context assertion issued by IAM Identity Center.
- **Data type** – [Arn](#), [String](#)
- **Value type** – Single-valued

`identitycenter:CredentialId`

This context key is a random ID for the identity-enhanced role credential and is used for logging only. Because this key value is unpredictable, we recommend that you do not use it for context assertions in policies.

- **Availability** – This key is included in the request context of an AWS STS `AssumeRole` API operation. The request context includes a context assertion issued by IAM Identity Center.
- **Data type** – [String](#)
- **Value type** – Single-valued

`identitycenter:InstanceArn`

This context key is the ARN of the instance of IAM Identity Center that issued the context assertion for the `identitystore:UserID`. You can use this key to determine whether the `identitystore:UserID` and context assertion came from an expected IAM Identity Center instance ARN.

- **Availability** – This key is included in the request context of an AWS STS `AssumeRole` API operation. The request context includes a context assertion issued by IAM Identity Center.
- **Data type** – [Arn](#), [String](#)
- **Value type** – Single-valued

Logging and monitoring in IAM Identity Center

As a best practice, you should monitor your organization to ensure that changes are logged. This helps you to ensure that any unexpected change can be investigated and unwanted changes can be

rolled back. AWS IAM Identity Center currently supports two AWS services that help you monitor your organization and the activity that happens within it.

Topics

- [Logging IAM Identity Center API calls with AWS CloudTrail](#)
- [Amazon EventBridge](#)
- [Logging AD sync and configurable AD sync errors](#)

Logging IAM Identity Center API calls with AWS CloudTrail

AWS IAM Identity Center is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in IAM Identity Center. CloudTrail captures API calls for IAM Identity Center as events. The calls captured include calls from the IAM Identity Center console and code calls to the IAM Identity Center API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for IAM Identity Center. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to IAM Identity Center, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

Topics

- [IAM Identity Center information in CloudTrail](#)
- [Understanding CloudTrail log file entries for IAM Identity Center](#)
- [Understanding IAM Identity Center sign-in events](#)

IAM Identity Center information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in IAM Identity Center, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing events with CloudTrail event history](#).

For an ongoing record of events in your AWS account, including events for IAM Identity Center, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events

from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple Regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

When CloudTrail logging is enabled in your AWS account, API calls made to IAM Identity Center actions are tracked in log files. IAM Identity Center records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

The following IAM Identity Center CloudTrail operations are supported:

Console API operations	Public API operations
AssociateDirectory	AttachManagedPolicyToPermissionSet
AssociateProfile	CreateAccountAssignment
BatchDeleteSession	CreateInstanceAccessControlAttributeConfiguration
BatchGetSession	CreatePermissionSet
CreateApplicationInstance	DeleteAccountAssignment
CreateApplicationInstanceCertificate	DeleteInlinePolicyFromPermissionSet
CreatePermissionSet	DeleteInstanceAccessControlAttributeConfiguration
CreateProfile	DeletePermissionSet

Console API operations	Public API operations
DeleteApplicationInstance	DescribeAccountAssignmentCreationStatus
DeleteApplicationInstanceCertificate	DescribeAccountAssignmentDeletionStatus
DeletePermissionsPolicy	DescribeInstanceAccessControlAttributeConfiguration
DeletePermissionSet	DescribePermissionSet
DeleteProfile	DescribePermissionSetProvisioningStatus
DescribePermissionsPolicies	DetachManagedPolicyFromPermissionSet
DisassociateDirectory	GetInlinePolicyForPermissionSet
DisassociateProfile	ListAccountAssignmentCreationStatus
GetApplicationInstance	ListAccountAssignmentDeletionStatus
GetApplicationTemplate	ListAccountAssignments
GetMfaDeviceManagementForDirectory	ListAccountsForProvisionedPermissionSet
GetPermissionSet	ListInstances
GetSSOStatus	ListManagedPoliciesInPermissionSet
ImportApplicationInstanceServiceProviderMetadata	ListPermissionSetProvisioningStatus

Console API operations	Public API operations
ListApplicationInstances	ListPermissionSets
ListApplicationInstanceCertificates	ListPermissionSetsProvisionedToAccount
ListApplicationTemplates	ListTagsForResource
ListDirectoryAssociations	ProvisionPermissionSet
ListPermissionSets	PutInlinePolicyToPermissionSet
ListProfileAssociations	TagResource
ListProfiles	UntagResource
ListSessions	UpdateInstanceAccessControlAttributeConfiguration
PutMfaDeviceManagementForDirectory	UpdatePermissionSet
PutPermissionsPolicy	
StartSSO	
UpdateApplicationInstanceActiveCertificate	
UpdateApplicationInstanceDisplayData	
UpdateApplicationInstanceServiceProviderConfiguration	
UpdateApplicationInstanceStatus	
UpdateApplicationInstanceResponseConfiguration	

Console API operations	Public API operations
UpdateApplicationInstanceResponseSchemaConfiguration	
UpdateApplicationInstanceSecurityConfiguration	
UpdateDirectoryAssociation	
UpdateProfile	

For more information about IAM Identity Center's public API operations, see the [IAM Identity Center API Reference Guide](#).

The following IAM Identity Center Identity Store CloudTrail operations are supported:

- AddMemberToGroup
- CompleteVirtualMfaDeviceRegistration
- CompleteWebAuthnDeviceRegistration
- CreateAlias
- CreateExternalIdPConfigurationForDirectory
- CreateGroup
- CreateUser
- DeleteExternalIdPConfigurationForDirectory
- DeleteGroup
- DeleteMfaDeviceForUser
- DeleteUser
- DescribeDirectory
- DescribeGroups
- DescribeUsers
- DisableExternalIdPConfigurationForDirectory
- DisableUser
- EnableExternalIdPConfigurationForDirectory

- EnableUser
- GetAWSSSPConfigurationForDirectory
- ListExternalIdPConfigurationsForDirectory
- ListGroupsForUser
- ListMembersInGroup
- ListMfaDevicesForUser
- PutMfaDeviceManagementForDirectory
- RemoveMemberFromGroup
- SearchGroups
- SearchUsers
- StartVirtualMfaDeviceRegistration
- StartWebAuthnDeviceRegistration
- UpdateExternalIdPConfigurationForDirectory
- UpdateGroup
- UpdateMfaDeviceForUser
- UpdatePassword
- UpdateUser
- VerifyEmail

The following IAM Identity Center OIDC CloudTrail actions are supported:

- CreateToken
- RegisterClient
- StartDeviceAuthorization

The following IAM Identity Center Portal CloudTrail actions are supported:

- Authenticate
- Federate
- ListApplications
- ListProfilesForApplication
- ListAccounts

- `ListAccountRoles`
- `GetRoleCredentials`
- `Logout`

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail `userIdentity` element](#).

Understanding CloudTrail log file entries for IAM Identity Center

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry for an administrator (`samadams@example.com`) that took place in the IAM Identity Center console:

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJAIENLMexample",
        "arn": "arn:aws:iam::08966example:user/samadams",
        "accountId": "08966example",
        "accessKeyId": "AKIAIIJM2K4example",
        "userName": "samadams"
      },
      "eventTime": "2017-11-29T22:39:43Z",
      "eventSource": "sso.amazonaws.com",
```

```

    "eventName": "DescribePermissionsPolicies",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36",
    "requestParameters": {
      "permissionSetId": "ps-79a0dde74b95ed05"
    },
    "responseElements": null,
    "requestID": "319ac6a1-d556-11e7-a34f-69a333106015",
    "eventID": "a93a952b-13dd-4ae5-a156-d3ad6220b071",
    "readOnly": true,
    "resources": [
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "08966example"
  }
]
}

```

The following example shows a CloudTrail log entry for an end-user (bobsmith@example.com) action that took place in the AWS access portal:

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "Unknown",
        "principalId": "example.com//
S-1-5-21-1122334455-3652759393-4233131409-1126",
        "accountId": "08966example",
        "userName": "bobsmith@example.com"
      },
      "eventTime": "2017-11-29T18:48:28Z",
      "eventSource": "sso.amazonaws.com",
      "eventName": "ListApplications",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "203.0.113.0",
      "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36",
      "requestParameters": null,

```

```

    "responseElements":null,
    "requestID":"de6c0435-ce4b-49c7-9bcc-bc5ed631ce04",
    "eventID":"e6e1f3df-9528-4c6d-a877-6b2b895d1f91",
    "eventType":"AwsApiCall",
    "recipientAccountId":"08966example"
  }
]
}

```

The following example shows a CloudTrail log entry for an end-user (bobsmith@example.com) action that took place in IAM Identity Center OIDC:

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "example.com//S-1-5-21-1122334455-3652759393-4233131409-1126",
    "accountId": "08966example",
    "userName": "bobsmith@example.com"
  },
  "eventTime": "2020-06-16T01:31:15Z",
  "eventSource": "sso.amazonaws.com",
  "eventName": "CreateToken",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36",
  "requestParameters": {
    "clientId": "clientid1234example",
    "clientSecret": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "grantType": "urn:ietf:params:oauth:grant-type:device_code",
    "deviceCode": "devicecode1234example"
  },
  "responseElements": {
    "accessToken": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "tokenType": "Bearer",
    "expiresIn": 28800,
    "refreshToken": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "idToken": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "eventID": "09a6e1a9-50e5-45c0-9f08-e6ef5089b262",
  "readOnly": false,
  "resources": [

```



```

    {
      "accountId": "08966example",
      "type": "IdentityStoreId",
      "ARN": "d-1234example"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "08966example"
}

```

Understanding IAM Identity Center sign-in events

AWS CloudTrail logs successful and unsuccessful sign-in events for all AWS IAM Identity Center identity sources. Native SSO and Active Directory (AD Connector and AWS Managed Microsoft AD) sourced identities will include additional sign-in events that are captured each time a user is prompted to solve a specific credential challenge or factor, as well as the status of that particular credential verification request. Only after a user has completed all required credential challenges will the user be signed in, which will result in a `UserAuthentication` event being logged.

The following table captures each of the IAM Identity Center sign-in CloudTrail event names, their purpose, and applicability to different identity sources.

Event name	Event purpose	Identity source applicability
<code>CredentialChallenge</code>	Used to notify that IAM Identity Center has requested the user to solve a specific credential challenge and specifies the <code>CredentialType</code> that was required (For example, <code>PASSWORD</code> or <code>TOTP</code>).	Native IAM Identity Center users, AD Connector, and AWS Managed Microsoft AD
<code>CredentialVerification</code>	Used to notify that the user has attempted to solve a specific <code>CredentialChallenge</code> request and specifies whether that	Native IAM Identity Center users, AD Connector, and AWS Managed Microsoft AD

Event name	Event purpose	Identity source applicability
	credential succeeded or failed.	
UserAuthentication	Used to notify that all authentication requirements the user was challenged with have been successfully completed and that the user was successfully signed in. <i>Users failing to successfully complete the required credential challenges will result in no UserAuthentication event being logged.</i>	All identity sources

The following table captures additional useful event data fields contained within specific sign-in CloudTrail events.

Event name	Event purpose	Sign-in event applicability	Example values
AuthWorkflowID	Used to correlate all events emitted across an entire sign-in sequence. For each user sign-in, multiple events may be emitted by IAM Identity Center.	CredentialChallenge, CredentialVerification, UserAuthentication	"AuthWorkflowID": "9de74b32-8362-4a01-a524-de21df59fd83"
CredentialType	Used to specify the credential or factor that was challenged. UserAuthentication	CredentialChallenge, CredentialVerification	CredentialType": "PASSWORD" or "CredentialType": "PASSWORD"

Event name	Event purpose	Sign-in event applicability	Example values
	Authentication events will include all of the <code>CredentialType</code> values that were successfully verified across the user's sign-in sequence.	Authentication, <code>UserAuthentication</code>	<code>,TOTP</code> (possible values include: <code>PASSWORD</code> , <code>TOTP</code> , <code>WEBAUTHN</code> , <code>EXTERNAL_IDP</code> , <code>RESYNC_TOTP</code>)
<code>DeviceEnrollmentRequired</code>	Used to specify that the user was required to register an MFA device during sign-in, and that the user successfully completed that request.	<code>UserAuthentication</code>	"DeviceEnrollmentRequired": "true"
<code>LoginTo</code>	Used to specify the redirect location following a successful sign-in sequence.	<code>UserAuthentication</code>	"LoginTo": "https://mydirectory.awsapps.com/start/...."

Example events for IAM Identity Center sign-in scenarios

The following examples show the expected sequence of CloudTrail events for different sign-in scenarios.

Topics

- [Successful sign-in when authenticating with only a password](#)
- [Successful sign-in when authenticating with an external identity provider](#)
- [Successful sign-in when authenticating with a password and a TOTP authenticator app](#)
- [Successful sign-in when authenticating with a password and forced MFA registration is required](#)
- [Failed sign-in when authenticating with only a password](#)

Successful sign-in when authenticating with only a password

The following sequence of events captures an example of a successful password only sign-in.

CredentialChallenge (Password)

```
{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"Unknown",
    "principalId":"111122223333",
    "arn":"",
    "accountId":"111122223333",
    "accessKeyId":"",
    "userName":"user1"
  },
  "eventTime":"2020-12-07T20:33:58Z",
  "eventSource":"signin.amazonaws.com",
  "eventName":"CredentialChallenge",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"203.0.113.0",
  "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters":null,
  "responseElements":null,
  "additionalEventData":{
    "AuthWorkflowID":"9de74b32-8362-4a01-a524-de21df59fd83",
    "CredentialType":"PASSWORD"
  },
  "requestID":"5be44ffb-6946-4f47-acaf-1adebd4afead",
  "eventID":"27ea7725-c1fd-4355-bdba-d0e628e0e604",
  "readOnly":false,
  "eventType":"AwsServiceEvent",
  "managementEvent":true,
  "eventCategory":"Management",
  "recipientAccountId":"111122223333",
  "serviceEventDetails":{
    "CredentialChallenge":"Success"
  }
}
```

Successful CredentialVerification (Password)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-07T20:34:09Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialVerification",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "9de74b32-8362-4a01-a524-de21df59fd83",
    "CredentialType": "PASSWORD"
  },
  "requestID": "f3cf52ad-fd3d-4889-8c15-f18d1a7c7393",
  "eventID": "c49640f6-0c8a-43d3-a6e0-900e3bb188d4",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "CredentialVerification": "Success"
  }
}
```

Successful UserAuthentication (Password Only)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
```

```

    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-07T20:34:09Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "UserAuthentication",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "9de74b32-8362-4a01-a524-de21df59fd83",
    "LoginTo": "https://d-1234567890.awsapps.com/start/?
state=QVlBQmVGMHFiS0wzWlp1SFgrR25BRnFobU5nQUlnQUJBQk5FWVhSaFVHeGhibVZUZEdGMFpWQmhjbUZ0QUFsUVpYS
BshlIc50BAA6ftz73M6LsflWDLf0xvi02K3wet9461C30f_iWdilx-
zv__4pSHf7mcUIs&wdc_csrf_token=srAzW1jK4GPYYoR452ruZ38DxEsDY9x81q1tVRSnno5pUjISvP7Tqzi0LiBLBUSx
east-1",
    "CredentialType": "PASSWORD"
  },
  "requestID": "f3cf52ad-fd3d-4889-8c15-f18d1a7c7393",
  "eventID": "e959a95a-2b33-478d-906c-4fe303e8a9f1",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "UserAuthentication": "Success"
  }
}

```

Successful sign-in when authenticating with an external identity provider

The following sequence of events captures an example of a successful sign-in when authenticated through the SAML protocol using an external identity provider.

Successful UserAuthentication (External Identity Provider)

```

{
  "eventVersion": "1.08",
  "userIdentity": {

```

```

    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": ""
  },
  "eventTime": "2020-12-07T20:34:09Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "UserAuthentication",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "9de74b32-8362-4a01-a524-de21df59fd83",
    "LoginTo": "https://d-1234567890.awsapps.com/start/?
state=QVlBQmVGMHFiS0wzWlp1SFgrR25BRnFobU5nQUlnQUJBQk5FWVhSaFVHeGhibVZUZEdGMFpWQmhjbUZ0QUFsUVpYS
BshlIc50BAA6ftz73M6LsflWDLf0xvi02K3wet9461C30f_iWdilx-
zv__4pSHf7mCUIs&wdc_csrf_token=srAzW1jK4GPYYoR452ruZ38DxEsDY9x81q1tVRSnno5pUjISvP7Tqzi0LiBLBUSx
east-1",
    "CredentialType": "EXTERNAL_IDP"
  },
  "requestID": "f3cf52ad-fd3d-4889-8c15-f18d1a7c7393",
  "eventID": "e959a95a-2b33-478d-906c-4fe303e8a9f1",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "UserAuthentication": "Success"
  }
}

```

Successful sign-in when authenticating with a password and a TOTP authenticator app

The following sequence of events captures an example where multi-factor authentication was required during sign-in and the user successfully signed in using a password and a TOTP authenticator app.

CredentialChallenge (Password)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-08T20:40:13Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialChallenge",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "303486b5-fce1-4d59-ba1d-eb3acb790729",
    "CredentialType": "PASSWORD"
  },
  "requestID": "e454ea66-1027-4d00-9912-09c0589649e1",
  "eventID": "d89cc0b5-a23a-4b88-843a-89329aaef2e",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "CredentialChallenge": "Success"
  }
}
```

Successful CredentialVerification (Password)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
```



```

    "accountId":"111122223333",
    "accessKeyId":"",
    "userName":"user1"
  },
  "eventTime":"2020-12-08T20:40:20Z",
  "eventSource":"signin.amazonaws.com",
  "eventName":"CredentialVerification",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"203.0.113.0",
  "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters":null,
  "responseElements":null,
  "additionalEventData":{
    "AuthWorkflowID":"303486b5-fce1-4d59-ba1d-eb3acb790729",
    "CredentialType":"PASSWORD"
  },
  "requestID":"92c4ac90-0d9b-452d-95d5-728487612f5e",
  "eventID":"4533fd49-6669-4d0b-b272-a0b2139309a8",
  "readOnly":false,
  "eventType":"AwsServiceEvent",
  "managementEvent":true,
  "eventCategory":"Management",
  "recipientAccountId":"111122223333",
  "serviceEventDetails":{
    "CredentialVerification":"Success"
  }
}

```

CredentialChallenge (TOTP)

```

{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"Unknown",
    "principalId":"111122223333",
    "arn":"",
    "accountId":"111122223333",
    "accessKeyId":"",
    "userName":"user1"
  },
  "eventTime":"2020-12-08T20:40:20Z",
  "eventSource":"signin.amazonaws.com",

```

```

    "eventName": "CredentialChallenge",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
      "AuthWorkflowID": "303486b5-fce1-4d59-ba1d-eb3acb790729",
      "CredentialType": "TOTP"
    },
    "requestID": "92c4ac90-0d9b-452d-95d5-728487612f5e",
    "eventID": "29202f08-f240-40cc-b789-c0cea8a27847",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
      "CredentialChallenge": "Success"
    }
  }
}

```

Successful CredentialVerification (TOTP)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-08T20:40:27Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialVerification",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,

```

```

"responseElements":null,
"additionalEventData":{
  "AuthWorkflowID":"303486b5-fce1-4d59-ba1d-eb3acb790729",
  "CredentialType":"TOTP"
},
"requestID":"c40a691f-eeb1-4352-b286-5e909f96f318",
"eventID":"e889ff1d-fcaf-454f-805d-7132cf2362a4",
"readOnly":false,
"eventType":"AwsServiceEvent",
"managementEvent":true,
"eventCategory":"Management",
"recipientAccountId":"111122223333",
"serviceEventDetails":{
  "CredentialVerification":"Success"
}
}

```

Successful UserAuthentication (Password + TOTP)

```

{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"Unknown",
    "principalId":"111122223333",
    "arn":"",
    "accountId":"111122223333",
    "accessKeyId":"",
    "userName":"user1"
  },
  "eventTime":"2020-12-08T20:40:27Z",
  "eventSource":"signin.amazonaws.com",
  "eventName":"UserAuthentication",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"203.0.113.0",
  "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters":null,
  "responseElements":null,
  "additionalEventData":{
    "AuthWorkflowID":"303486b5-fce1-4d59-ba1d-eb3acb790729",
    "LoginTo":"https://d-1234567890.awsapps.com/start/?state
\u003dQV1BQmVLeFhWeDRmZFJmMmxHcWYwdzhZck5RQU1nQUJBQk5FWVhSaFVHeGhibVZUZEdGMFpWQmhjbUZ0QUFsUVpYS
\u0026auth_code

```

```

\u003d11Fir1mCVJ-4Y5UY6RI10UCXvRePCHd6195xvYg1rwo1Pj7B-7UGIG1YUUVe31Nkzd7ihxKn6DMdnFf00108qc3RF
Sx-pjBXXG_jUcvBk_UILdGytV4o1u97h42B-
TA_6uwdmJiw1dcCz_Rv44d_BS0Pku1W-5LVJy1oeP1H0FPPMeheyuk5Uy48d5of9-c\u0026wdc_csrf_token
\u003dNMLui44guoVnxRd0qu2tYJIdyyFPX6SDRNTspIScfMM0AgFbho1nvvCaxPTghHbgHCRIXdffFtzH0sL1ow419Bobn
\u0026organization\u003dd-9067230c03\u0026region\u003dus-east-1",
  "CredentialType": "PASSWORD, TOTP"
},
"requestID": "c40a691f-eeb1-4352-b286-5e909f96f318",
"eventID": "7a8c8725-db2f-488d-a43e-788dc6c73a4a",
"readOnly": false,
"eventType": "AwsServiceEvent",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "111122223333",
"serviceEventDetails": {
  "UserAuthentication": "Success"
}
}

```

Successful sign-in when authenticating with a password and forced MFA registration is required

The following sequence of events captures an example of a successful password sign in, but the user was required and successfully completed registering an MFA device before completing their sign-in.

CredentialChallenge (Password)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-09T01:24:02Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialChallenge",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",

```

```

    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
      "AuthWorkflowID": "76d8a26d-ad9c-41a4-90c3-d607cdd7155c",
      "CredentialType": "PASSWORD"
    },
    "requestID": "321f4b13-42b5-4005-a0f7-826cad26d159",
    "eventID": "8c707b0f-e45a-4a9c-bee2-ff68638d2f1b",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
      "CredentialChallenge": "Success"
    }
  }
}

```

Successful CredentialVerification (Password)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-09T01:24:09Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialVerification",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "76d8a26d-ad9c-41a4-90c3-d607cdd7155c",

```

```

    "CredentialType": "PASSWORD"
  },
  "requestID": "12b57efa-0a92-4479-91a3-5b6641817c21",
  "eventID": "783b0c89-7142-4942-8b84-6ee0de1b992e",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "CredentialVerification": "Success"
  }
}

```

Successful UserAuthentication (Password + MFA Registration Required)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-09T01:24:14Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "UserAuthentication",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "76d8a26d-ad9c-41a4-90c3-d607cdd7155c",
    "LoginTo": "https://d-1234567890.awsapps.com/start/?state
\u003dQVlBQmVGQ3VqdHF5aW9CUDdrNXRTVTJUaWNNQUlnQUJBQk5FWVhSaFVHeGhibVZUZEdGMFpWQmhjbUZ0QUFsUVpYS
\u0026auth_code
\u003d11eZ80S_maUsZ7ABETjeQhyWfvIHYz52rgR28sYAKN1oEk2G07czrwzXvE9HL1N2K9De8LyBEV83SFeDQfrWpkwXf
FJyJqkoGrt_w6rm_MpAn0uyrVq8udY_EgU3fh0L3QWvWiquYnDPMYPmmy_qkZgR9rz__BI
\u0026wdc_csrf_token

```

```

\u003dJih9U62o5LQDtYLNqCK8a6xj0gJg5BRWq2t175y8vAmwZhAqrgrgbxXat2M646UZGp93krw7WYQdHIgi50YI9QSc
\u003dd-9067230c03\u0026region\u003dus-east-1",
  "CredentialType":"PASSWORD",
  "DeviceEnrollmentRequired":"true"
},
"requestID":"74d24604-a365-4237-8c4a-350795494b92",
"eventID":"a15bf257-7f37-46c0-b67c-fea5fa6166be",
"readOnly":false,
"eventType":"AwsServiceEvent",
"managementEvent":true,
"eventCategory":"Management",
"recipientAccountId":"111122223333",
"serviceEventDetails":{
  "UserAuthentication":"Success"
}
}

```

Failed sign-in when authenticating with only a password

The following sequence of events captures an example of a failed password only sign-in.

CredentialChallenge (Password)

```

{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"Unknown",
    "principalId":"111122223333",
    "arn":"",
    "accountId":"111122223333",
    "accessKeyId":"",
    "userName":"user1"
  },
  "eventTime":"2020-12-08T18:56:15Z",
  "eventSource":"signin.amazonaws.com",
  "eventName":"CredentialChallenge",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"203.0.113.0",
  "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters":null,
  "responseElements":null,
  "additionalEventData":{
    "AuthWorkflowID":"adbf67c4-8188-4e2b-8527-fe539e328fa7",

```

```

    "CredentialType": "PASSWORD"
  },
  "requestID": "f54848ea-b1aa-402f-bf0d-a54561a2ffcc",
  "eventID": "d96f1d6c-dbd9-4a0b-9a45-6a2b66078c78",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "CredentialChallenge": "Success"
  }
}

```

Failed CredentialVerification (Password)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "111122223333",
    "arn": "",
    "accountId": "111122223333",
    "accessKeyId": "",
    "userName": "user1"
  },
  "eventTime": "2020-12-08T18:56:21Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "CredentialVerification",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "AuthWorkflowID": "adbf67c4-8188-4e2b-8527-fe539e328fa7",
    "CredentialType": "PASSWORD"
  },
  "requestID": "04528c82-a678-4a1f-a56d-ea2c6445a72a",
  "eventID": "9160fe06-fc2a-474f-9b78-000ee067a09d",
  "readOnly": false,
  "eventType": "AwsServiceEvent",

```



```
"managementEvent":true,
"eventCategory":"Management",
"recipientAccountId":"111122223333",
"serviceEventDetails":{"
  "CredentialVerification":"Failure"
}
}
```

Amazon EventBridge

IAM Identity Center can work with Amazon EventBridge to raise events when administrator-specified actions occur in an organization. For example, because of the sensitivity of such actions, most administrators would want to be warned every time someone creates a new account in the organization or when an administrator of a member account attempts to leave the organization. You can configure EventBridge rules that look for these actions and then send the generated events to administrator-defined targets. Targets can be an Amazon SNS topic that emails or text messages its subscribers. You could also create an AWS Lambda function that logs the details of the action for your later review.

To learn more about EventBridge, including how to configure and enable it, see the [Amazon EventBridge User Guide](#).

Logging AD sync and configurable AD sync errors

You can enable logging on your Active Directory (AD) sync and configurable AD sync configurations to receive logs with information about errors that can occur during the sync process. With these logs, you can monitor if there is an issue with your AD sync and configurable AD sync and take action if applicable. You can send your logs to an Amazon CloudWatch Logs log group, an Amazon Simple Storage Service (Amazon S3) bucket, or an Amazon Data Firehose with cross account delivery supported for Amazon S3 buckets and Firehose.

For more information about limitations, permissions, and vended logs, see [Enabling logging from AWS services](#).

Note

You are charged for logging. For more information, see [Vended Logs](#) on the [Amazon CloudWatch Pricing](#) page.

To enable AD sync and configurable AD sync error logs

1. Sign in to the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage logs**.
4. Choose **Add log delivery** and one of the following destination types.
 - a. Choose **To Amazon CloudWatch Logs**. Then choose or enter the destination log group.
 - b. Choose **To Amazon S3**. Then choose or enter the destination bucket.
 - c. Choose **To Firehose**. Then choose or enter the destination delivery stream.
5. Choose **Submit**.

To disable AD sync and configurable AD sync error logs

1. Sign in to the [IAM Identity Center console](#).
2. Choose **Settings**.
3. On the **Settings** page, choose the **Identity source** tab, choose **Actions**, and then choose **Manage logs**.
4. Choose **Remove** for the destination that you want to remove.
5. Choose **Submit**.

AD sync and configurable AD sync error log fields

See the following list for possible error log fields.

`sync_profile_name`

The name of the sync profile.

`error_code`

The error code that represents what type of error has occurred.

`error_message`

A message that contains detailed information about the error that occurred.

sync_source

The sync source is where entities are being synced from. For IAM Identity Center, this is an Active Directory (AD) managed by AWS Directory Service. The sync source contains the domain and ARN of the directory affected.

sync_target

The sync target is the destination where entities are being saved. For IAM Identity Center, this is an Identity Store. The sync target contains the Identity Store ARN affected.

source_entity_id

A unique identifier for the entity that is causing the error. For IAM Identity Center, this is the SID of the entity.

source_entity_type

The type of entity causing the error. The value can be USER or GROUP.

eventTimestamp

The timestamp when the error occurred.

AD sync and configurable AD sync error log examples

Example 1: An error log for an expired password for an AD directory

```
{
  "sync_profile_name": "EXAMPLE-PROFILE-NAME",
  "error" : {
    "error_code": "InvalidDirectoryCredentials",
    "error_message": "The password for your AD directory has expired. Please reset the password to allow Identity Sync to access the directory."
  },
  "sync_source": {
    "arn": "arn:aws:ds:us-east-1:123456789:directory/d-123456",
    "domain": "EXAMPLE.com"
  },
  "eventTimestamp": "1683355579981"
}
```

Example 2: An error log for a user with a non-unique username

```
{
  "sync_profile_name": "EXAMPLE-PROFILE-NAME",
  "error" : {
    "error_code": "ConflictError",
    "error_message": "The source entity has a username conflict with the sync
target. Please verify that the source identity has a unique username in the target."
  },
  "sync_source": {
    "arn": "arn:aws:ds:us-east-1:111122223333:directory/d-123456",
    "domain": "EXAMPLE.com"
  },
  "sync_target": {
    "arn": "arn:aws:identitystore::111122223333:identitystore/d-123456"
  },
  "source_entity_id": "SID-1234",
  "source_entity_type": "USER",
  "eventTimestamp": "1683355579981"
}
```

Compliance validation for IAM Identity Center

Third-party auditors assess the security and compliance of AWS services such as AWS IAM Identity Center as part of multiple AWS compliance programs.

To learn whether an AWS service is within the scope of specific compliance programs, see [AWS services in Scope by Compliance Program](#) and choose the compliance program that you are interested in. For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

Note

Not all AWS services are HIPAA eligible. For more information, see the [HIPAA Eligible Services Reference](#).

- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [AWS Customer Compliance Guides](#) – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see [Security Hub controls reference](#).
- [Amazon GuardDuty](#) – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- [AWS Audit Manager](#) – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Supported compliance standards

IAM Identity Center has undergone auditing for the following standards and is eligible for use as part of solutions for which you need to obtain compliance certification.



AWS has expanded its Health Insurance Portability and Accountability Act (HIPAA) compliance program to include IAM Identity Center as a [HIPAA eligible service](#).

AWS offers a [HIPAA-focused whitepaper](#) for customers who want to learn more about how they can use AWS services to process and store health information. For more information, see [HIPAA compliance](#).



The Information Security Registered Assessors Program (IRAP) enables Australian Government customers to ensure that appropriate compliance controls are in place and determine the appropriate responsibility model for addressing the requirements of the Australian Government Information Security Manual (ISM) produced by the Australian Cyber Security Centre (ACSC). For more information, see [IRAP Resources](#).



IAM Identity Center has an Attestation of Compliance for Payment Card Industry (PCI) Data Security Standard (DSS) version 3.2 at Service Provider Level 1.

Customers who use AWS products and services to store, process, or transmit cardholder data can use the following identity sources in IAM Identity Center to manage their own PCI DSS compliance certification:

- Active Directory
- External identity provider

The IAM Identity Center identity source is currently not compliant with PCI DSS.

For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS level 1](#).



System & Organization Control (SOC) Reports are independent, third-party examination reports that demonstrate how IAM Identity Center achieves key compliance controls and objectives. These reports help you and your auditors to understand how controls support operations and compliance. There are three types of SOC reports:

- AWS SOC 1 Report - [Download with AWS Artifact](#)
- AWS SOC 2: Security, Availability, & Confidentiality Report - [Download with AWS Artifact](#)
- [AWS SOC 3: Security, Availability, & Confidentiality Report](#)

IAM Identity Center is in scope for AWS SOC 1, SOC 2, and SOC 3 reports. For more information, see [SOC Compliance](#).

Resilience in IAM Identity Center

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS global infrastructure](#).

To learn more about AWS IAM Identity Center resiliency, see [Resiliency design and Regional behavior](#).

Infrastructure security in IAM Identity Center

As a managed service, AWS IAM Identity Center is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see [AWS Cloud](#)

[Security](#). To design your AWS environment using the best practices for infrastructure security, see [Infrastructure Protection](#) in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access IAM Identity Center through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Tagging AWS IAM Identity Center resources

A *tag* is a custom attribute label that you add to an AWS resource to make it easier to identify, organize, and search for resources. Each tag has two parts:

- A *tag key* (for example, `CostCenter`, `Environment`, or `Project`). Tag keys can be up to 128 characters in length and are case sensitive.
- A *tag value* (for example, `111122223333` or `Production`). Tag values can be up to 256 characters in length, and like tag keys, are case sensitive. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. Omitting the tag value is the same as using an empty string.

Tags help you identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related. For example, you can assign the same tag to a specific permission set in your instance of IAM Identity Center. For more information about tagging strategies, see [Tagging AWS Resources](#) in the *AWS General Reference Guide* and [Tagging Best Practices](#).

In addition to identifying, organizing, and tracking your AWS resources with tags, you can use tags in IAM policies to help control who can view and interact with your resources. To learn more about using tags to control access, see [Controlling access to AWS resources using tags](#) in the *IAM User Guide*. For example, you can allow a user to update an IAM Identity Center permission set, but only if the IAM Identity Center permission set has an `owner` tag with a value of that user's name.

Currently, you can apply tags to permission sets only. You can't apply tags to the corresponding roles that IAM Identity Center creates in AWS accounts. You can use the IAM Identity Center console, AWS CLI or the IAM Identity Center APIs to add, edit, or delete tags for a permission set.

The following sections provide more information about tags for IAM Identity Center.

Topics

- [Tag restrictions](#)
- [Manage tags by using the IAM Identity Center console](#)
- [AWS CLI examples](#)
- [Manage tags using the IAM Identity Center API](#)

Tag restrictions

The following basic restrictions apply to tags on IAM Identity Center resources:

- The maximum number of tags that you can assign to a resource is 50.
- The maximum key length is 128 Unicode characters.
- The maximum value length is 256 Unicode characters.
- Valid characters for a tag key and value are:

a-z, A-Z, 0-9, space, and the following characters: `_` `.` `:` `/` `=` `+` `-` and `@`

- Keys and values are case sensitive.
- Don't use `aws :` as a prefix for keys; it's reserved for AWS use

Manage tags by using the IAM Identity Center console

You can use the IAM Identity Center console to add, edit, and remove tags that are associated with your instance or permission sets.

To manage permission sets tags for an IAM Identity Center console

1. Open the [IAM Identity Center console](#).
2. Choose **Permission sets**.
3. Choose the name of the permission set that has the tags you want to manage.
4. On the **Permissions** tab, under **Tags**, do one of the following, and then proceed to the next step:
 - a. If tags are already assigned for this permission set, choose **Edit tags**.
 - b. If no tags are assigned to this permission set, choose **Add tags**.
5. For each new tag, type the values in the **Key** and **Value (optional)** columns. When you are finished, choose **Save changes**.

To remove a tag, choose the **X** in the **Remove** column next to the tag that you want to remove.

To manage tags for an instance of IAM Identity Center

1. Open the [IAM Identity Center console](#).

2. Choose **Settings**.
3. Choose the **Tags** tab.
4. For each tag, type the values in the **Key** and **Value (optional)** fields. When you are finished, choose the **Add new tag** button.

To remove a tag, choose the **Remove** button next to the tag that you want to remove.

AWS CLI examples

The AWS CLI provides commands that you can use to manage the tags that you assign to your permission set.

Assigning tags

Use the following commands to assign tags to your permission set.

Example tag-resource Command for a permission set

Assign tags to a permission set by using [tag-resource](#) within the `sso` set of commands:

```
$ aws sso-admin tag-resource \  
> --instance-arn sso-instance-arn \  
> --resource-arn sso-resource-arn \  
> --tags Stage=Test
```

This command includes the following parameters:

- `instance-arn` – The Amazon Resource Name (ARN) of the IAM Identity Center instance under which the operation will run.
- `resource-arn` – The ARN of the resource with the tags to be listed.
- `tags` – The key-value pairs of the tags.

To assign multiple tags at once, specify them in a comma-separated list:

```
$ aws sso-admin tag-resource \  
> --instance-arn sso-instance-arn \  
> --resource-arn sso-resource-arn \  
> --tags Stage=Test,Environment=Dev
```

```
> --tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Viewing tags

Use the following commands to view the tags that you have assigned to your permission set.

Example `list-tags-for-resource` Command for a permission set

View the tags that are assigned to a permission set by using [list-tags-for-resource](#) within the sso set of commands:

```
$ aws sso-admin list-tags-for-resource --resource-arn sso-resource-arn
```

Removing tags

Use the following commands to remove tags from a permission set.

Example `untag-resource` Command for a permission set

Remove tags from a permission set by using [untag-resource](#) within the sso set of commands:

```
$ aws sso-admin untag-resource \  
> --instance-arn sso-instance-arn \  
> --resource-arn sso-resource-arn \  
> --tag-keys Stage CostCenter Owner
```

For the `--tag-keys` parameter, specify one or more tag keys, and do not include the tag values.

Applying tags when you create a permission set

Use the following commands to assign tags at the moment you create a permission set.

Example `create-permission-set` Command with tags

When you create a permission set by using the [create-permission-set](#) command, you can specify tags with the `--tags` parameter:

```
$ aws sso-admin create-permission-set \  
> --instance-arn sso-instance-arn \  
> --name permission=set-name \  
> --tags Stage=Test, CostCenter=80432, Owner=SysEng
```

```
> --tags Stage=Test, CostCenter=80432, Owner=SysEng
```

Manage tags using the IAM Identity Center API

You can use the following actions in the IAM Identity Center API to manage the tags for your permission set.

API actions for IAM Identity Center instance tags

Use the following API actions to assign, view, and remove tags for a permission set or instance of IAM Identity Center.

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)
- [CreatePermissionSet](#)
- [CreateInstance](#)

Integrating AWS CLI with IAM Identity Center

AWS Command Line Interface (CLI) version 2 integration with IAM Identity Center simplifies the sign-in process. Developers can sign in directly to the AWS CLI using the same Active Directory or IAM Identity Center credentials that they normally use to sign in to IAM Identity Center, and access their assigned accounts and roles. For example, after an administrator configures IAM Identity Center to use Active Directory for authentication, a developer can sign into the AWS CLI directly using their Active Directory credentials.

AWS CLI integration with IAM Identity Center offers the following benefits:

- Enterprises can enable their developers to sign in using credentials from IAM Identity Center or Active Directory by connecting IAM Identity Center to their Active Directory using AWS Directory Service.
- Developers can sign in from the CLI for faster access.
- Developers can list and switch between accounts and roles to which they have assigned access.
- Developers can generate and save named role profiles in their CLI configuration automatically and reference them in the CLI to run commands in desired accounts and roles.
- The CLI manages short-term credentials automatically so developers can start in and stay in the CLI securely without interruption, and run long running scripts.

How to integrate AWS CLI with IAM Identity Center

To use the AWS CLI integration with IAM Identity Center, you need to download, install, and configure AWS Command Line Interface version 2. For detailed steps on how to download and integrate the AWS CLI with IAM Identity Center, see [Configuring the AWS CLI to use IAM Identity Center](#) in the *AWS Command Line Interface User Guide*.

AWS IAM Identity Center Region availability

You can enable IAM Identity Center in most AWS Regions and it's available to users globally. This global availability makes it easier for you to configure user access to multiple AWS accounts and applications. When your users sign in to the [AWS access portal](#), they can select the AWS account to which they have permissions, and then access the AWS Management Console. For a full list of the AWS Regions that IAM Identity Center supports, see [IAM Identity Center endpoints and quotas](#).

IAM Identity Center Region data

When you enable IAM Identity Center, all the data that you configure in IAM Identity Center is stored in the Region where you configured it. This data includes directory configurations, permission sets, application instances, and user assignments to AWS account applications. If you are using the IAM Identity Center identity store, all users and groups that you create in IAM Identity Center are also stored in the same Region.

Choosing your Region

We recommend that you install IAM Identity Center in a Region that you intend to keep available for users, not a Region that you might need to disable. See [Considerations for choosing an AWS Region](#).

If you enable an organization instance of IAM Identity Center in the management account of your AWS organization in one Region and later decide to switch to a different Region, you must first [delete your current IAM Identity Center instance](#). Switching to a different Region also changes the URL for the AWS access portal, and you must reconfigure all permission sets and assignments.

Cross-Region calls

IAM Identity Center uses [Amazon Simple Email Service \(Amazon SES\)](#) to send emails to end users when they attempt to sign-in with one-time password (OTP) as a second authentication factor. These emails are also sent for certain identity and credential management events, such as when the user is invited to set up an initial password, to verify an email address, and reset their password. Amazon SES is available in a subset of AWS Regions that IAM Identity Center supports.

IAM Identity Center calls Amazon SES local endpoints when Amazon SES is available locally in an AWS Region. When Amazon SES isn't available locally, IAM Identity Center calls Amazon SES endpoints in a different AWS Region, as indicated in the following table.

IAM Identity Center Region code	IAM Identity Center Region name	Amazon SES Region code	Amazon SES Region name
ap-east-1	Asia Pacific (Hong Kong)	ap-northeast-2	Asia Pacific (Seoul)
ap-south-2	Asia Pacific (Hyderabad)	ap-south-1	Asia Pacific (Mumbai)
ap-southeast-4	Asia Pacific (Melbourne)	ap-southeast-2	Asia Pacific (Sydney)
ca-west-1	Canada West (Calgary)	ca-central-1	Canada (Central)
eu-south-2	Europe (Spain)	eu-west-3	Europe (Paris)
eu-central-2	Europe (Zurich)	eu-central-1	Europe (Frankfurt)
me-central-1	Middle East (UAE)	eu-central-1	Europe (Frankfurt)
us-gov-east-1	AWS GovCloud (US-East)	us-gov-west-1	AWS GovCloud (US-West)

In these cross-Region calls, IAM Identity Center might send the following user attributes:

- Email address
- First name
- Last name
- Account in AWS Organizations
- AWS access portal URL
- Username
- Directory ID
- User ID

Managing IAM Identity Center in an opt-in Region (Region that is disabled by default)

Most AWS Regions are enabled for operations in all AWS services by default. Those Regions are automatically activated for use with IAM Identity Center. The following AWS Regions are opt-in Regions and you must enable them if you want to use IAM Identity Center:

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Melbourne)
- Canada West (Calgary)
- Europe (Milan)
- Europe (Spain)
- Europe (Zurich)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- Middle East (UAE)

When you enable IAM Identity Center for a management account in an opt-in AWS Region, the following IAM Identity Center metadata for any member accounts is stored in the Region.


- Account ID
- Account name
- Account email
- Amazon Resource Names (ARNs) of the IAM roles that IAM Identity Center creates in the member account

Disabling an AWS Region where IAM Identity Center is enabled

If you disable an AWS Region in which IAM Identity Center is installed, IAM Identity Center is also disabled. After IAM Identity Center is disabled in a Region, users in that Region won't have single

sign-on access to AWS accounts and applications. AWS retains the data in your IAM Identity Center configuration for at least 10 days. If you re-enable the AWS Region within this time frame, your IAM Identity Center configuration data will still be available in the Region.

To re-enable IAM Identity Center in opt-in AWS Regions, you must re-enable the Region. Because IAM Identity Center must reprocess all paused events again, re-enabling IAM Identity Center might take some time.

 **Note**

IAM Identity Center can manage access only to the AWS accounts that are enabled for use in an AWS Region. To manage access across all accounts in your organization, enable IAM Identity Center in the management account in an AWS Region that's automatically activated for use with IAM Identity Center.

For more information about enabling and disabling AWS Regions, see [Managing AWS Regions](#) in the *AWS General Reference*.

Quotas and limits in IAM Identity Center

The following tables describe quotas within IAM Identity Center. Quota increase requests must come from a management or delegated administrator account. To increase a quota, see [Requesting a quota increase](#).

Note

We recommend using the AWS CLI and APIs if you have more than 50,000 users, 10,000 groups, or 500 permission sets. For more information about the CLI, see [Integrating AWS CLI with IAM Identity Center](#). For more information about APIs, see [Welcome to the IAM Identity Center API Reference](#).

Application quotas

Resource	Default quota	Can be increased
File size of service provider SAML certificates (in PEM format)	2 KB	No
SAML assertion limit	50,000 characters	No
File size limit of the IdP certificate uploaded to IAM Identity Center	2500 (UTF-8) characters	No
Access scopes per application	25	No

AWS account quotas

Resource	Default quota	Can be increased
Number of permission sets allowed in IAM Identity Center	2000	Yes
Number of provisioned permission sets allowed per AWS account	250	Yes
Number of inline policies per permission set	1	No
Number of AWS managed and customer managed policies per permission set	20 ¹	No
Maximum size of inline policy per permission set	32,768 bytes. Maximum size of non-white space characters in the inline policy per permission set is 10,240 bytes.	No
Number of IAM roles (permission sets) in the AWS account that can be updated at a time	1	No

¹AWS Identity and Access Management (IAM) sets a quota of 10 managed policies per role. To take advantage of this quota, request an increase to the IAM quota *Managed policies attached to an IAM role* in the Service Quotas console for each AWS account where you want to deploy the permission set.

Note

[Manage AWS accounts with permission sets](#) are provisioned in AWS accounts as IAM roles, or use existing IAM roles in AWS accounts, and therefore follow IAM quotas. For more information about quotas that are associated with IAM roles, see [IAM and STS quotas](#).

Active Directory quotas

Resource	Default quota	Can be increased
Number of connected directories that you can have at a time	1	No

IAM Identity Center identity store quotas

Resource	Default quota	Can be increased
Number of users supported in IAM Identity Center	100000	Yes
Number of groups supported in IAM Identity Center	100000	No
Number of unique groups that can be used to evaluate the permissions for a user	1000	No

IAM Identity Center throttle limits

Resource	Default quota
IAM Identity Center APIs	IAM Identity Center APIs have a collective throttle maximum of 20 transactions per

Resource	Default quota
	second (TPS). The CreateAccountAssignment has a maximum rate of 10 outstanding async calls. These quotas cannot be changed.

Additional quotas

Resource	Default quota	Can be increased
Total number of AWS accounts or applications that can be configured *	3000	Yes
Total number of instances of IAM Identity Center per account	1	No
Total number of trusted token issuers	10	No

* Up to 3000 AWS accounts or applications (total combined) are supported. For example, you might configure 2750 accounts and 250 applications, resulting in a total of 3000 accounts and applications.

Troubleshooting IAM Identity Center issues

The following can help you troubleshoot some common issues you might encounter while setting up or using the IAM Identity Center console.

Issues creating an account instance of IAM Identity Center

Several restrictions might apply when creating an account instance of IAM Identity Center. If you're unable to create an account instance through the IAM Identity Center console, or the setup experience of a supported AWS managed application, verify the following use cases:

- Check other AWS Regions in the AWS account in which you are attempting to create the account instance. You are limited to one instance of IAM Identity Center per AWS account. To enable the application, either switch to the AWS Region with the instance of IAM Identity Center or switch to an account without an instance of IAM Identity Center.
- If your organization enabled IAM Identity Center before September 14, 2023, your administrator might need to opt-in to account instance creation. Work with your administrator to enable account instance creation from the IAM Identity Center console in the management account.
- Your administrator might have created a Service Control Policy to limit creation of account instances of IAM Identity Center. Work with your administrator add your account to the allow list.

You receive an error when you attempt to view the list of cloud applications that are preconfigured to work with IAM Identity Center

This following error occurs when you have a policy that allows `sso:ListApplications` but not other IAM Identity Center APIs. Update your policy to address this error.

The `ListApplications` permission authorizes multiple APIs:

- The `ListApplications` API.
- An internal API similar to the `ListApplicationProviders` API used in the IAM Identity Center console.

To help resolve duplication, the internal API now also authorizes using the `ListApplicationProviders` action. To allow the public `ListApplications` API but deny the internal API, your policy must include a statement denying the `ListApplicationProviders` action:

```
"Statement": [  
  {  
    "Effect": "Deny",  
    "Action": "sso:ListApplicationProviders",  
    "Resource": "*"  
  },  
  {  
    "Effect": "Allow",  
    "Action": "sso:ListApplications",  
    "Resource": "<instanceArn>" // (or "*" for all instances)  
  }  
]
```

To allow the internal API but deny `ListApplications`, the policy needs to allow only `ListApplicationProviders`. The `ListApplications` API is denied if not explicitly allowed.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": "sso:ListApplicationProviders",  
    "Resource": "*"  
  }  
]
```

When your policies are updated, contact AWS Support to have this proactive measure removed.

Issues regarding contents of SAML assertions created by IAM Identity Center

IAM Identity Center provides a web-based debug experience for the SAML assertions created and sent by IAM Identity Center, including attributes within these assertions, when accessing AWS

accounts and SAML applications from the AWS access portal. To see the details of a SAML assertion that IAM Identity Center generates, use the following steps.

1. Sign in to the AWS access portal.
2. While you are signed into the portal, hold the **Shift** key down, choose the application tile, and then release the **Shift** key.
3. Examine the information on the page titled **You are now in administrator mode**. To keep this information for future reference, choose **Copy XML**, and paste the contents elsewhere.
4. Choose **Send to <application>** to continue. This option sends the assertion to the service provider.

Note

Some browser configurations and operating systems may not support this procedure. This procedure has been tested on Windows 10 using Firefox, Chrome, and Edge browsers.

Specific users fail to synchronize into IAM Identity Center from an external SCIM provider

If SCIM synchronization succeeds for a subset of users configured in your IdP for provisioning into IAM Identity Center but fails for others, you might see an error similar to 'Request is unparsable, syntactically incorrect, or violates schema' from your identity provider. You may also see detailed provisioning failure messages in AWS CloudTrail.

This issue often indicates that the user in your IdP is configured in a way that IAM Identity Center does not support. Full details of the IAM Identity Center SCIM implementation, including the specifications of required, optional, and forbidden parameters and operations for user objects, can be found in the [IAM Identity Center SCIM Implementation Developer Guide](#). The *SCIM Developer Guide* should be considered authoritative for information around SCIM requirements. However, the following are a couple of common reasons for this error:

1. The user object in the IdP lacks a first (given) name, a last (family) name, and/or a display name.

- **Solution:** Add a first (given), last (family), and display name for the user object. In addition, ensure that the SCIM provisioning mappings for user objects at your IdP are configured to send nonempty values for all of these attributes.
2. More than one value for a single attribute is being sent for the user (also known as “multi-value attributes”). For example, the user may have both a work and a home phone number specified in the IdP, or multiple emails or physical addresses, and your IdP is configured to try to synchronize multiple or all values for that attribute.
 - **Solution options:**
 - i. Update your SCIM provisioning mappings for user objects at your IdP to send only a single value for a given attribute. For example, configure a mapping that sends only the work phone number for each user.
 - ii. If the additional attributes can safely be removed from the user object at the IdP, you can remove the additional values, leaving either one or zero values set for that attribute for the user.
 - iii. If the attribute is not needed for any actions in AWS, remove the mapping for that attribute from the SCIM provisioning mappings for user objects at your IdP.
 3. Your IdP is trying to match users in the target (IAM Identity Center, in this case) based on multiple attributes. Since user names are guaranteed unique within a given IAM Identity Center instance, you only need to specify `username` as the attribute used for matching.
 - **Solution:** Ensure that your SCIM configuration in your IdP is using only a single attribute for matching with users in IAM Identity Center. For example, mapping `username` or `userPrincipalName` in the IdP to the `username` attribute in SCIM for provisioning to IAM Identity Center will be correct and sufficient for most implementations.

Users can't sign in when their user name is in UPN format

Users might not be able to sign in to the AWS access portal based on the format they use to enter in their user name on the sign in page. For the most part, users can sign in to the user portal using either their plain user name, their down-level logon name (DOMAIN\UserName) or their UPN logon name (UserName@Corp.Example.com). The exception to this is when IAM Identity Center is using a connected directory that has been enabled with MFA and the verification mode has been set to either **Context-aware** or **Always-on**. In this scenario, users must sign in with their down-level logon name (DOMAIN\UserName). For more information, see [Multi-factor authentication for Identity](#)

[Center users](#). For general information about user name formats used to sign in to Active Directory, see [User Name Formats](#) on the Microsoft documentation website.

I get a 'Cannot perform the operation on the protected role' error when modifying an IAM role

When reviewing IAM Roles in an account, you may notice role names beginning with 'AWSReservedSSO_'. These are the roles which the IAM Identity Center service has created in the account, and they came from assigning a permission set to the account. Attempting to modify these roles from within the IAM console will result in the following error:

```
'Cannot perform the operation on the protected role 'AWSReservedSSO_RoleName_Here' - this role is only modifiable by AWS'
```

These roles can only be modified from the IAM Identity Center Administrator console, which is in the management account of AWS Organizations. Once modified, you can then push the changes down to the AWS accounts that it is assigned to.

Directory users cannot reset their password

When a directory user resets their password using the **Forgot Password?** option during sign-in of the AWS access portal, their new password must adhere to the default password policy as described in [Password requirements when managing identities in IAM Identity Center](#).

If a user enters a password that adheres to the policy and then receives the error `We couldn't update your password`, check to see if AWS CloudTrail recorded the failure. This can be done by searching in the Event History console of CloudTrail using the following filter:

```
"UpdatePassword"
```

If the message states the following, then you may need to contact support:

```
"errorCode": "InternalFailure",  
  "errorMessage": "An unknown error occurred"
```

Another possible cause of this issue is in the naming convention that was applied to the user name value. Naming conventions must follow specific patterns such as 'surname.givenName'. However,

some user names can be quite long, or contain special characters, and this can cause characters to be dropped in the API call, thereby resulting in an error. You may want to attempt a password reset with a test user in the same manner to verify if this is the case.

If the issue persists, contact the [AWS Support Center](#).

My user is referenced in a permission set but can't access the assigned accounts or applications

This issue can occur if you're using System for Cross-domain Identity Management (SCIM) for Automatic Provisioning with an external identity provider. Specifically, when a user, or the group the user was a member of, is deleted then re-created using the same user name (for users) or name (for groups) in the identity provider, a new unique internal identifier is created for the new user or group in IAM Identity Center. However, IAM Identity Center still has a reference to the old identifier in its permission database, such that the name of the user or group still appears in the UI, but access fails. This is because the underlying user or group ID to which the UI refers no longer exists.

To restore AWS account access in this case, you can remove access for the old user or group from the AWS account(s) where it was originally assigned, and then reassign access back to the user or group. This updates the permission set with the correct identifier for the new user or group. Similarly, to restore application access, you can remove access for the user or group from the assigned users list for that application, then add the user or group back again.

You can also check to see if AWS CloudTrail recorded the failure by searching your CloudTrail logs for SCIM synchronization events that reference the name of the user or group in question.

I can't get my application from the application catalog configured correctly

If you added an application from the application catalog in IAM Identity Center, be aware that each service provider provides their own detailed documentation. You can access this information from the **Configuration** tab for the application in the IAM Identity Center console.

If the problem is related to setting up the trust between the service provider's application and IAM Identity Center, make sure to check the instruction manual for troubleshooting steps.

Error 'An unexpected error has occurred' when a user tries to sign in using an external identity provider

This error may occur for multiple reasons, but one common reason is a mis-match between the user information carried in the SAML request, and the information for the user in IAM Identity Center.

In order for an IAM Identity Center user to sign in successfully when using an external IdP as the identity source, the following must be true:

- The SAML nameID format (configured at your identity provider) must be 'email'
- The nameID value must be a properly (RFC2822)-formatted string (user@domain.com)
- The nameID value must exactly match the user name of an existing user in IAM Identity Center (it doesn't matter if the email address in IAM Identity Center matches or not – the inbound match is based on username)
- The IAM Identity Center implementation of SAML 2.0 federation supports only 1 assertion in the SAML response between the identity provider and IAM Identity Center. It does not support encrypted SAML assertions.
- The following statements apply if [Attributes for access control](#) is enabled in your IAM Identity Center account:
 - The number of attributes mapped in the SAML request must be 50 or less.
 - The SAML request must not contain multi-valued attributes.
 - The SAML request must not contain multiple attributes with the same name.
 - The attribute must not contain structured XML as the value.
 - The Name format must be a SAML specified format, not generic format.

Note

IAM Identity Center does not perform “just in time” creation of users or groups for new users or groups via SAML federation. This means that the user must be pre-created in IAM Identity Center, either manually or via automatic provisioning, in order to sign in to IAM Identity Center.

This error can also occur when the Assertion Consumer Service (ACS) endpoint configured in your identity provider does not match the ACS URL provided by your IAM Identity Center instance. Ensure that these two values match exactly.

Additionally, you can troubleshoot external identity provider sign-in failures further by going to AWS CloudTrail and filtering on the event name **ExternalIdPDirectoryLogin**.

Error 'Attributes for access control failed to enable'

This error may occur if the user enabling ABAC does not have the `iam:UpdateAssumeRolePolicy` permissions required to enable [Attributes for access control](#).

I get a 'Browser not supported' message when I attempt to register a device for MFA

WebAuthn is currently supported in Google Chrome, Mozilla Firefox, Microsoft Edge and Apple Safari web browsers, as well as Windows 10 and Android platforms. Some components of WebAuthn support may be varied, such as platform authenticator support across macOS and iOS browsers. If users attempt to register WebAuthn devices on an unsupported browser or platform, they will see certain options greyed out that are not supported, or they will receive an error that all supported methods are not supported. In these cases, please refer to [FIDO2: Web Authentication \(WebAuthn\)](#) for more information about browser/platform support. For more information about WebAuthn in IAM Identity Center, see [FIDO2 authenticators](#).

Active Directory "Domain Users" group does not properly sync into IAM Identity Center

The Active Directory Domain Users group is the default "primary group" for AD user objects. Active Directory primary groups and their memberships cannot be read by IAM Identity Center. When assigning access to IAM Identity Center resources or applications, use groups other than the Domain Users group (or other groups assigned as primary groups) to have group membership properly reflected in the IAM Identity Center identity store.

Invalid MFA credentials error

This error can occur when a user attempts to sign in to IAM Identity Center using an account from an external identity provider (for example, Okta or Microsoft Entra ID) before their account is fully

provisioned to IAM Identity Center using the SCIM protocol. After the user account is provisioned to IAM Identity Center, this issue should be resolved. Confirm that the account has been provisioned to IAM Identity Center. If not, check the provisioning logs in the external identity provider.

I get a 'An unexpected error has occurred' message when I attempt to register or sign in using an authenticator app

Time-based one-time password (TOTP) systems, such as those used by IAM Identity Center in combination with code-based authenticator apps, rely on time synchronization between the client and the server. Ensure that the device where your authenticator app is installed is correctly synchronized to a reliable time source, or manually set the time on your device to match a reliable source, such as NIST (<https://www.time.gov/>) or other local/regional equivalents.

I get an 'It's not you, it's us' error when attempting to sign in to IAM Identity Center

This error indicates there is a setup problem with your instance of IAM Identity Center or the external identity provider (IdP) IAM Identity Center is using as its identity source. We recommend you verify the following:

- Verify the date and time settings on the device you're using to sign in. We recommend that you set the date and time to be set automatically. If that is not available, we recommend syncing your date and time to a known Network Time Protocol (NTP) server.
- Verify that the IdP certificate uploaded to IAM Identity Center is the same as what was provided by your IdP. You can check the certificate from the IAM Identity Center console by navigating to **Settings**. In the **Identity Source** tab choose **Action** and then choose **Manage Authentication**. If the IdP and IAM Identity Center certificates do not match, import a new certificate to IAM Identity Center.
- Ensure the NameID format in your identity provider's metadata file is the following:
 - `urn:oasis:name:tc:SAML:1.1:nameid-format:emailAddress`
- If you are using AD Connector from AWS Directory Service as your identity provider, verify that the credentials for the service account are correct and have not expired. See [Update your AD Connector service account credentials in AWS Directory Service](#) for more information.

My users are not receiving emails from IAM Identity Center

All emails sent by the IAM Identity Center service will come from either the address `no-reply@signin.aws` or `no-reply@login.awsapps.com`. Your mail system must be configured so that it accepts emails from these sender email addresses and doesn't handle them as junk or spam.

Error: You can't delete/modify/remove/assign access to permission sets provisioned in the management account

This message indicates that the [Delegated administration](#) feature has been enabled and that the operation you previously attempted can only be successfully performed by someone who has management account permissions in AWS Organizations. To resolve this issue, sign in as a user who has these permissions and try performing the task again or assign this task to someone who has the correct permissions. For more information, see [Register a member account](#).

Error: Session token not found or invalid

This error can occur when a client, such as a web browser, AWS Toolkit, or AWS CLI, tries to use a session that's revoked or invalidated on the server side. To correct this issue, return to the client application or website and try again, including logging in again if prompted. This might sometimes require you to also cancel pending requests, such as a pending connection attempt from the AWS Toolkit within your IDE.

Document history

The following table describes important additions to the AWS IAM Identity Center documentation. We also update the documentation frequently to address the feedback that you send us.

- **Latest major documentation update:** September 23, 2022

Change	Description	Date
Updates to the "What is IAM Identity Center?" topic	Updated the content that describes the benefits and capabilities of IAM Identity Center.	August 19, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	July 12, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	June 27, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	May 17, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	April 30, 2024

	ntityContext AWS managed policy.	
Updates for AWS managed policy	Updated permissions for the AWSSSOMasterAccountAdministrator AWS managed policy.	April 26, 2024
Updates for AWS managed policy	Updated permissions for the AWSSSOMemberAccountAdministrator AWS managed policy.	April 26, 2024
Updates for AWS managed policy	Updated permissions for the AWSSS0ReadOnly AWS managed policy.	April 26, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	April 26, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	April 24, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	April 19, 2024

Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	April 11, 2024
Updates for AWS managed policy	Updated permissions for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	November 26, 2023
New AWS managed policy topic	Added details for the AWSIAMIdentityCenterAllowListForIdentityContext AWS managed policy.	November 15, 2023
Enhanced guidance for getting started with IAM Identity Center	Added new content for getting started with IAM Identity Center and creating an administrative user	September 23, 2022
Updated users and groups in the Identity Center API Reference	This update includes references to the new Create, Update and Delete APIs in the Identity Center API Reference Guide.	August 31, 2022

[AWS Single Sign-On \(AWS SSO\) renamed to AWS IAM Identity Center](#)

AWS introduces AWS IAM Identity Center. IAM Identity Center expands the capabilities of AWS Identity and Access Management (IAM) to help you centrally manage account and access to applications for your workforce users. IAM Identity Center features include application assignments, multi-account permissions, and an AWS access portal.

July 26, 2022

[Support for permissions boundaries and customer managed policies in permission sets](#)

Added content for using AWS managed and customer managed AWS Identity and Access Management (IAM) policies with permission sets.

July 14, 2022

[Support for manually enabled AWS Regions](#)

Added content for using IAM Identity Center in manually enabled Regions.

June 15, 2022

[Updates for AWS managed policies](#)

Updated permissions for the `AWSSS0ServiceRolePolicy` AWS managed policy.

May 11, 2022

[Support for delegated administration](#)

Added content for the delegated administration feature.

May 11, 2022

Updates for AWS managed policies	Updated permissions for the <code>AWSSSOMasterAccountAdministrator</code> , <code>AWSSSOMemberAccountAdministrator</code> , and <code>AWSSS0ReadOnly</code> AWS managed policies.	April 28, 2022
Support for configurable AD sync	Added content for the configurable AD sync feature.	April 14, 2022
New AWS managed policy topic	Added details for the <code>AWSSSOMasterAccountAdministrator</code> AWS managed policy.	August 4, 2021
Updates for quotas	Adjustments to quota tables.	December 21, 2020
New example policies	Added new customer managed policy examples and updates to the permissions required section.	December 21, 2020
Support for attribute-based access control (ABAC)	Added content for ABAC feature.	November 24, 2020
Support for MFA forced enrollment	Updates to require users to enroll an MFA device at sign-in.	November 23, 2020
Support for WebAuthn	Added content for new WebAuthn feature.	November 20, 2020
Support for Ping Identity	Added content to integrate with Ping Identity products as a supported external identity provider.	October 26, 2020

Support for OneLogin	Added content to integrate with OneLogin as a supported external identity provider.	July 31, 2020
Support for Okta	Added content to integrate with Okta as a supported external identity provider.	May 28, 2020
Support for external identity providers	Changed references from directory to identity source, added content to support external identity providers.	November 26, 2019
New MFA settings	Removed two-step verification topic and added new MFA topic in its place.	October 24, 2019
New setting to add two-step verification	Added content on how to enable two-step verification for users.	January 16, 2019
Support for session duration on AWS accounts	Added content on how to set the session duration for an AWS account.	October 30, 2018
New option to use Identity Center directory	Added content for choosing either Identity Center directory or connecting to an existing directory in Active Directory.	October 17, 2018
Support for relay state and session duration on applications	Added content about relay state and session duration for applications.	October 10, 2018

Additional support for new applications	Added 4me, BambooHR, Bonusly, Citrix ShareFile, ClickTime, Convo, Deputy, Deskpro, Dome9, DruvalnSync, Egnyte, Engagedly, Expensify, Freshdesk, IdeaScale, Igloo, Jitbit, Kudos, LiquidFiles, Lucidchart, PurelyHR, Samanage, ScreenSteps, Sli.do, SmartSheet, Syncplicity, TalentLMS, Trello, UserVoice, Zoho, OpsGenie, DigiCert, WeekDone, ProdPad, and UserEcho to the application catalog.	August 3, 2018
Support for multi-account access to management accounts	Added content about how to delegate multi-account access to users in a management account.	July 9, 2018
Support for new applications	Added DocuSign, Keeper Security, and SugarCRM to the application catalog.	March 16, 2018
Get temporary credentials for CLI access	Added information about how to get temporary credentials to run AWS CLI commands.	February 22, 2018
New guide	This is the first release of the IAM Identity Center User Guide.	December 7, 2017

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.