User Guide

AWS IAM Identity Center

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IAM Identity Center: User Guide

Copyright © 2026 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
Getting started ... 4
IAM Identity Center prerequisites and considerations ... 6
Choosing an AWS Region ... 7
Using IAM Identity Center for applications only ... 13
IAM roles created by IAM Identity Center ... 14
IAM Identity Center and AWS Organizations ... 15
IAM Identity Center instances ... 16
AWS account types that can enable IAM Identity Center ... 16
Organization instances of IAM Identity Center ... 18
Account instances of IAM Identity Center ... 19
Delete your IAM Identity Center instance ... 24
Enable IAM Identity Center ... 26
To enable an instance of IAM Identity Center ... 27
Confirm your identity sources ... 29
Update firewalls and gateways ... 32
Considerations for allowlisting domains and URL endpoints ... 34
Identity source tutorials ... 35
Active Directory ... 36
CyberArk ... 38
Prerequisites ... 39
SCIM considerations ... 39
Step 1: Enable provisioning in IAM Identity Center ... 40
Step 2: Configure provisioning in CyberArk ... 41
(Optional) Step 3: Configure user attributes in CyberArk for access control (ABAC) in IAM Identity Center ... 42
(Optional) Passing attributes for access control ... 42
Google Workspace ... 43
Considerations ... 44
Step 1: Google Workspace: Configure the SAML application ... 45

Step 2: IAM Identity Center and Google Workspace: Change the IAM Identity Center identity source and setup Google Workspace as an SAML identity provider ... 45
Step 3: Google Workspace: Enable the apps ... 47
Step 4: IAM Identity Center: Set up IAM Identity Center automatic provisioning ... 47
Step 5: Google Workspace: Configure auto provisioning ... 48
Passing attributes for access control - *Optional* ... 50
Assign access to AWS accounts ... 50
Next steps ... 53
Troubleshooting ... 53
JumpCloud ... 54
Prerequisites ... 55
SCIM considerations ... 55
Step 1: Enable provisioning in IAM Identity Center ... 56
Step 2: Configure provisioning in JumpCloud ... 56
(Optional) Step 3: Configure user attributes in JumpCloud for access control in IAM Identity Center ... 57
(Optional) Passing attributes for access control ... 58
Microsoft Entra ID ... 59
Prerequisites ... 59
Considerations ... 59
Step 1: Prepare your Microsoft tenant ... 61
Step 2: Prepare your AWS account ... 63
Step 3: Configure and test your SAML connection ... 66
Step 4: Configure and test your SCIM synchronization ... 70
Step 5: Configure ABAC - *Optional* ... 74
Assign access to AWS accounts ... 75
Microsoft Entra ID configuration for access to additional Regions ... 77
Troubleshooting ... 78
Okta ... 80
Considerations ... 81
Step 1: Okta: Obtain the SAML metadata from your Okta account ... 82
Step 2: IAM Identity Center: Configure Okta as the identity source for IAM Identity Center ... 83
Step 3: IAM Identity Center and Okta: Provision Okta users ... 84
Step 4: Okta: Synchronize users from Okta with IAM Identity Center ... 85
Passing attributes for access control - *Optional* ... 87

Assign access to AWS accounts ... 88
Okta configuration for access to additional Regions ... 90
Next steps ... 90
Troubleshooting ... 91
OneLogin ... 93
Prerequisites ... 94
Step 1: Enable provisioning in IAM Identity Center ... 94
Step 2: Configure provisioning in OneLogin ... 95
(Optional) Step 3: Configure user attributes in OneLogin for access control in IAM Identity Center ... 96
(Optional) Passing attributes for access control ... 97
Troubleshooting ... 98
Ping Identity ... 99
PingFederate ... 99
PingOne ... 106
Identity Center directory ... 112
Video tutorials ... 118
Set up your workforce ... 119
Users, groups, and provisioning ... 119
Username and email address uniqueness ... 119
Groups ... 119
User and group provisioning ... 120
User and group deprovisioning ... 120
Manage your identity source ... 120
Considerations for changing your identity source ... 121
Change your identity source ... 127
User and group attributes in IAM Identity Center ... 128
External identity providers ... 130
Microsoft AD directory ... 147
Manage users in the Identity Center directory ... 167
Provisioning when users are in IAM Identity Center ... 168
Changing your identity source ... 168
Add users ... 168
Add groups ... 171
Add users to groups ... 172
Delete groups ... 173

Delete users ... 174
Remove users from groups ... 176
Edit user properties ... 176
Workforce access ... 178
Understanding authentication sessions ... 178
Types of authentication sessions ... 179
Ways to end user sessions ... 180
What happens to user access when you end a session ... 181
Single logout ... 184
Best practices for session management ... 184
Configure session duration ... 184
User interactive sessions ... 185
User background sessions ... 186
Extended sessions for Kiro ... 188
End active sessions for workforce users ... 188
Considerations for identity sources, the AWS CLI, and AWS SDKs ... 190
Disable user access ... 193
Deny user access ... 194
Managing access for users in the Identity Center directory ... 196
Password management ... 196
MFA ... 196
Setting up user passwords ... 197
Multi-factor authentication ... 200
Application access ... 213
AWS managed applications ... 214
Controlling access to applications ... 214
Sharing identity information ... 215
Constraining the use of AWS managed applications ... 216
Applications that you can use with IAM Identity Center ... 217
Setting up IAM Identity Center to test AWS managed applications ... 221
Viewing and changing application details ... 227
Disabling an AWS managed application ... 228
Enabling identity-enhanced console sessions ... 228
Customer managed applications ... 231
SAML 2.0 and OAuth 2.0 applications ... 232
SAML 2.0 application setup ... 237

Trusted identity propagation ... 241
Benefits of trusted identity propagation ... 241
Enabling trusted identity propagation ... 241
How trusted identity propagation works ... 242
Prerequisites and considerations ... 243
Use cases ... 245
Authorization services ... 270
Setting up your own OAuth 2.0 application ... 280
Set up customer managed applications ... 281
Specify trusted applications ... 285
Trusted token issuer ... 286
Rotate certificates ... 300
Considerations before rotating a certificate ... 301
Rotate an IAM Identity Center certificate ... 301
Certificate expiration status indicators ... 304
Understand application properties ... 304
Application start URL ... 304
Relay state ... 305
Session duration ... 306
Assign user access to applications ... 306
Remove user access to applications ... 307
Map attributes ... 308
AWS account access ... 309
AWS account types ... 309
Assigning AWS account access ... 311
End-user experience ... 311
Enforcing and limiting access ... 312
Delegating and enforcing access ... 312
Limiting access to the identity store from member accounts ... 313
Delegated administration ... 313
Best practices ... 314
Prerequisites ... 319
Register a member account ... 319
Deregister a member account ... 321
View delegated administrator accounts ... 322
Temporary elevated access ... 322

Single sign-on access to AWS accounts ... 323
Assign user or group access to AWS accounts ... 324
Remove user and group access to an AWS account ... 327
Revoke an active permission set session ... 327
Delegate who can assign single sign-on access ... 329
Permission sets ... 331
Create a permission set that applies least-privilege permissions ... 331
Predefined permissions ... 333
Custom permissions ... 334
Create, manage, and delete permission sets ... 336
Configure permission set properties ... 349
Referencing permission sets ... 355
Recommendations to avoid access disruptions ... 357
Custom trust policy example ... 358
Attribute-based access control ... 359
Benefits ... 360
Checklist: Configuring ABAC in AWS using IAM Identity Center ... 361
Attributes for access control ... 363
Service-linked roles ... 371
AWS access portal ... 372
Getting started with the AWS access portal ... 372
Configure the AWS access portal ... 373
What you can configure ... 373
Activating the AWS access portal ... 373
Customizing the AWS access portal URL ... 374
Confirm users can sign in to the AWS access portal ... 375
Use the AWS access portal ... 376
How to use the AWS access portal ... 376
Signing in to the AWS access portal ... 377
Resetting your user password ... 379
AWS CLI and AWS SDK access ... 381
Creating shortcut links ... 386
Registering your device for MFA ... 389
Ending your active session ... 391
Using IAM Identity Center across multiple AWS Regions ... 393
Benefits of multi-Region support ... 393

Prerequisites and considerations ... 393
Choosing an additional Region ... 394
Replicate IAM Identity Center to an additional Region ... 395
Step 1: Create a replica key ... 395
Step 2: Add the Region ... 395
Step 3: Update external IdP setup ... 397
Step 4: Confirm firewall and gateway allow-lists ... 397
Step 5: Provide information to your users ... 397
Region changes beyond adding the first Region ... 398
What data is replicated? ... 398
Workforce access through an additional Region ... 399
User sessions across multiple AWS Regions ... 399
AWS access portal endpoints in the primary and additional AWS Regions ... 400
ACS endpoints in the primary and additional AWS Regions ... 401
Using AWS managed applications without multiple ACS URLs ... 403
Failover to an additional Region for AWS account access ... 403
AWS account access resiliency without multiple ACS URLs ... 404
Deploying and managing applications across multiple AWS Regions ... 405
Deploying and managing AWS managed applications across multiple AWS Regions ... 405
Deploying and managing customer managed applications across multiple AWS Regions ... 407
Remove a Region from IAM Identity Center ... 407
Step 1: Update external IdP configuration ... 407
Step 2: Remove the Region ... 408
Step 3: Delete the replica key ... 409
Service APIs supported in additional AWS Regions ... 409
CloudTrail events in the primary and additional Regions ... 410
IAM Identity Center use cases in the primary and additional Regions ... 410
Resiliency design and Regional behavior ... 413
Designed for availability ... 414
Set up emergency access to the AWS Management Console ... 414
Summary of emergency access configuration ... 415
How to design your critical operations roles ... 416
How to plan your access model ... 417
How to design emergency role, account, and group mapping ... 418
How to create your emergency access configuration ... 418
Emergency preparation tasks ... 420

Emergency failover process ... 420
Return to normal operations ... 421
One-time setup of a direct IAM federation application in Okta ... 421
One-time setup of a direct IAM federation application with ADFS ... 424
Security ... 431
Identity and access management for IAM Identity Center ... 432
Authentication ... 432
Access control ... 432
Overview of managing access ... 433
Identity-based policy examples ... 437
Resource-based policy example ... 445
AWS managed policies ... 448
Using service-linked roles ... 467
IAM Identity Center console and API authorization ... 475
API actions after November 2023 ... 475
API actions after October 2020 ... 476
AWS STS condition keys for IAM Identity Center ... 478
UserId ... 479
IdentityStoreArn ... 480
ApplicationArn ... 480
CredentialId ... 480
InstanceArn ... 481
Logging and monitoring ... 481
Logging IAM Identity Center API calls with AWS CloudTrail ... 481
Logging IAM Identity Center SCIM with AWS CloudTrail ... 521
Amazon EventBridge ... 528
Logging configurable AD sync errors ... 528
Compliance validation ... 531
Supported compliance standards ... 531
Resilience ... 533
Infrastructure security ... 533
Data protection ... 535
Encryption in transit ... 535
Data privacy ... 536
Data retention ... 536
Encryption at rest ... 536

IAM Identity Center encryption context ... 537
Using encryption context to control access to your customer managed key ... 538
Monitoring your encryption keys for IAM Identity Center ... 538
AWS managed applications' storage, encryption, and deletion of IAM Identity Center identity attributes ... 539
Implementing customer managed KMS keys ... 539
Baseline KMS key policy ... 549
Advanced KMS key policy statements ... 554
Tagging resources ... 563
Tag restrictions ... 564
Manage tags with the console ... 564
AWS CLI examples ... 565
Assigning tags ... 565
Viewing tags ... 566
Removing tags ... 566
Applying tags when you create a permission set ... 566
API actions ... 567
Integrating AWS CLI with IAM Identity Center ... 568
How to integrate AWS CLI with IAM Identity Center ... 568
Considerations for Private Access ... 569
Quotas ... 570
Application quotas ... 570
AWS account quotas ... 571
Active Directory quotas ... 572
IAM Identity Center identity store quotas ... 572
IAM Identity Center throttle limits ... 572
OIDC service request quotas ... 574
Additional quotas ... 575
Troubleshooting ... 577
Issues creating an account instance of IAM Identity Center ... 577
You receive an error when you attempt to view the list of cloud applications that are preconfigured to work with IAM Identity Center ... 577
Issues regarding contents of SAML assertions created by IAM Identity Center ... 578
Specific users fail to synchronize into IAM Identity Center from an external SCIM provider ... 579
Duplicate user or group error when provisioning users or groups with an external identity provider ... 581

Users can’t sign in when their user name is in UPN format 582
I get a ‘Cannot perform the operation on the protected role' error when modifying an IAM role 582
Directory users cannot reset their password 582
My user is referenced in a permission set but can’t access the assigned accounts or applications 583
I cannot get my application from the application catalog configured correctly 584
Error 'An unexpected error has occurred' when a user tries to sign in using an external identity provider 584
Error 'Attributes for access control failed to enable' 585
I get a 'Browser not supported' message when I attempt to register a device for MFA 585
Active Directory “Domain Users” group does not properly sync into IAM Identity Center 585
Invalid MFA credentials error 586
I get a 'An unexpected error has occurred' message when I attempt to register or sign in using an authenticator app 586
I get an 'It's not you, it is us' error when attempting to sign in to IAM Identity Center 586
My users are not receiving emails from IAM Identity Center 587
Error: You cannot delete/modify/remove/assign access to permission sets provisioned in the management account 587
Error: Session token not found or invalid 587
Group members from trusted domains do not sync to IAM Identity Center 587
Troubleshoot customer managed keys in AWS IAM Identity Center 588
Access Denied: KMS Decrypt Permission Issue 588
AWS managed application login failures with a customer managed KMS key enabled in IAM Identity Center 589
AWS managed application installation and/or user assignment failures with a customer managed KMS key enabled in IAM Identity Center 589
KMS Permissions Issue: Configuring Customer Managed Key with AWS IAM Identity Center 589
AWS access portal login failures with a customer managed KMS key enabled in IAM Identity Center 590
Troubleshoot multi-Region setup in AWS IAM Identity Center 590
The Region I want to replicate my IAM Identity Center instance to is not available in the IAM Identity Center console 590
AWS managed application sign-in failures in an additional Region 590
Document history 592

AWS Glossary 602

What is IAM Identity Center?

AWS IAM Identity Center is the AWS solution for connecting your workforce users to AWS managed applications such as Kiro and Amazon Quick, 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 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 Kiro 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 Quick 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 IAM Identity Center identity source tutorials. If you don't have an existing identity provider, you can create and manage users directly in IAM Identity Center.

Two instance types

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 is 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 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.

Multi-Region access to AWS accounts and applications

When you replicate your IAM Identity Center instance to additional Regions, your workforce can access their assigned AWS accounts and applications through all enabled Regions, and they can deploy AWS managed applications in each enabled Region.

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`
- CloudFormation resources containing `AWS::SSO` prefixes
- Service-linked role containing `AWSServiceRoleForSSO`
- Console URLs containing `sso` and `singlesignon`
- Documentation URLs containing `singlesignon`

Getting started with IAM Identity Center

The following outlines how you can get started with IAM Identity Center.

1. **Enable IAM Identity Center**

   When you enable IAM Identity Center, you choose between two types of IAM Identity Center instances. These types are: *organization instances* (recommended) and *account instances*. To learn more about the different capabilities of these instance types, see organization and account instances of IAM Identity Center.

   > **Note**
   >
   > After IAM Identity Center is enabled, you can sign in and open the IAM Identity Center console by doing either of the following:
   >
   > - **Organization instance** - Sign in to AWS using credentials with administrative permissions in the management account.
   > - **Account instance** - Sign in to AWS using credentials with administrative permissions in the AWS account where IAM Identity Center is enabled.

2. **Connect your identity source to IAM Identity Center**

   In IAM Identity Center console, confirm the identity source that you want to use. See the following for identity sources:

   - **External identity provider** - If you have an existing identity provider to manage your workforce users, you can connect it to IAM Identity Center. For more information about how to configure commonly used identity providers to work with IAM Identity Center, see IAM Identity Center identity source tutorials.
   - **Active Directory** - If you are using Active Directory to manage your workforce users, you can connect it to IAM Identity Center. For more information, see Using Active Directory as an identity source.
   - **IAM Identity Center** - Alternatively, you can create and manage users and groups directly in IAM Identity Center.

> **Note**
>
> Presently, you must use an external identity provider as the identity source to take advantage of a multi-Region setup with your IAM Identity Center. For more information about the benefits of this setup, see *Using IAM Identity Center across multiple AWS Regions*.

3. **Set up user access to AWS accounts (organization instance only)**

   If you're using an organization instance of IAM Identity Center, you can assign user or group access to AWS accounts, using permission sets to grant your users access to AWS accounts and resources.

4. **Set up user access to applications**

   With IAM Identity Center, you can grant users access to two types of applications:

   a. **AWS managed applications**

      - You can use IAM Identity Center with AWS managed applications like Amazon Q Business, AWS CLI, and Amazon Redshift. For more information, see AWS managed applications and Integrating AWS CLI with IAM Identity Center.

   b. **Customer managed applications**

      - You can integrate either of the following types of customer managed applications with IAM Identity Center:
        - Applications listed in IAM Identity Center catalog
        - Your custom applications
      - After configuring your application, you can assign your users access to the application.

5. **Provide your users with sign-in instructions for the AWS access portal**

   The AWS access portal is a web portal that provides your users with seamless access to all their assigned applications, AWS accounts, or both. New users in IAM Identity Center must activate their user credentials before they can sign in to the AWS access portal.

   For information about how to sign in to the AWS access portal, see Sign in to the AWS access portal in the *AWS Sign-In User Guide*. To learn about the sign-in process for the AWS access portal, see Signing in to the AWS access portal.

IAM Identity Center prerequisites and considerations

You can use IAM Identity Center for access to AWS managed applications only, AWS accounts only, or both. If you are using IAM federation to manage access to AWS accounts, you can continue to do so while using IAM Identity Center for application access.

Before enabling IAM Identity Center, consider the following:

- AWS Region

  You first enable IAM Identity Center in a single, supported Region for each instance of IAM Identity Center. If you want to use IAM Identity Center for single-sign on access to AWS accounts, the Region must be accessible by all of the users in your organization. If you plan to use IAM Identity Center for application access, be aware that some AWS managed applications, such as Amazon SageMaker AI, can operate only in the Regions they support. Also, most AWS managed applications require IAM Identity Center to be available in the same Region as the application. This can be achieved by co-locating them in the same Region, or when supported, by replicating the IAM Identity Center instance to the desired deployment Region of an AWS managed application. For more information, see Considerations for choosing an AWS Region.

- Application access only

  You can use IAM Identity Center only for user access to applications such as Kiro, using your existing identity provider. For more information, see Using IAM Identity Center for user access to applications only.

  > **Note**
  >
  > Access to application resources is managed independently by the application owner.

- Quota for IAM roles

  IAM Identity Center creates IAM roles to give users permissions to account resources. For more information, see IAM roles created by IAM Identity Center.

- 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 do not have to. 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. For more information, see IAM Identity Center and AWS Organizations.

IAM Identity Center web interfaces, including the access portal and the IAM Identity Center console, are intended to be accessed by humans through supported web browsers. Compatible browsers include the latest three versions of Microsoft Edge, Mozilla Firefox, Google Chrome, and Apple Safari. Accessing these endpoints using non-browser based paths is not supported. For programmatic access to IAM Identity Center services, we recommend using the documented APIs available in the IAM Identity Center and Identity Store API reference guides.

Considerations for choosing an AWS Region

You can enable IAM Identity Center in a single, supported AWS Region of your choice and it is available to users globally. This global availability makes it easier for you to configure user access to multiple AWS accounts and applications. Following are key considerations for choosing an AWS Region.

- **Geographical location of your users** – 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 AI.
- **Opt-in Regions (Regions that are disabled by default)** – An opt-in Region is an AWS Region that is disabled by default. To use an opt-in Region, you must enable it. For more information, see Managing IAM Identity Center in an opt-in Region.
- **Replicating IAM Identity Center to additional Regions** – If you plan to replicate IAM Identity Center to additional AWS Regions, you must choose a Region enabled by default. For more information, see *Using IAM Identity Center across multiple AWS Regions*.
- **Choosing deployment Regions for AWS managed applications** – AWS managed applications can operate only in the AWS Regions in which they are available. Many AWS managed applications can also operate only in a Region where IAM Identity Center is enabled or replicated to (primary or additional Region). To confirm if your IAM Identity Center instance supports replication to additional Regions, see *Using IAM Identity Center across multiple AWS Regions*. If replication is not an option, consider enabling IAM Identity Center in the Region where you plan to use AWS managed applications.
- **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 your self-managed directory in Active Directory (AD) as the identity source, its home Region must match the AWS Region in which you enabled IAM Identity Center.
- **Cross-Region emails with Amazon Simple Email Service** – In some Regions, IAM Identity Center may call Amazon Simple Email Service (Amazon SES) 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, see Cross-Region emails with Amazon SES.
- **AWS Control Tower** – If you're enabling an organization instance of IAM Identity Center from AWS Control Tower, the instance will be created in the same Region as the AWS Control Tower landing zone.

Topics

- IAM Identity Center Region data storage and operations
- Switching AWS Regions
- Disabling an AWS Region where IAM Identity Center is enabled

IAM Identity Center Region data storage and operations

Learn how IAM Identity Center handles data storage and operations across AWS Regions.

Understand how IAM Identity Center stores data

When you enable IAM Identity Center, all the data that you configure in IAM Identity Center is stored in the Region where you enabled 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. If you replicate your IAM Identity Center instance to additional Regions, IAM Identity Center automatically replicates users, groups, permission sets and their assignments, and other metadata and configuration to those Regions.

Cross-Region emails with Amazon SES

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-east-2	Asia Pacific (Taipei)	ap-northeast-1	Asia Pacific (Tokyo)
ap-south-2	Asia Pacific (Hyderabad)	ap-south-1	Asia Pacific (Mumbai)
ap-southeast-4	Asia Pacific (Melbourne)	ap-southeast-2	Asia Pacific (Sydney)
ap-southeast-5	Asia Pacific (Malaysia)	ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-6	Asia Pacific (New Zealand)	ap-southeast-2	Asia Pacific (Sydney)
ap-southeast-7	Asia Pacific (Thailand)	ap-northeast-3	Asia Pacific (Osaka)
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)
mx-central-1	Mexico (Central)	us-east-2	US East (Ohio)
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, but you must enable the following opt-in Regions if you want to use IAM Identity Center:

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Taipei)
- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Melbourne)
- Asia Pacific (Malaysia)
- Asia Pacific (New Zealand)
- Asia Pacific (Thailand)
- Canada West (Calgary)
- Europe (Milan)
- Europe (Spain)
- Europe (Zurich)
- Israel (Tel Aviv)
- Mexico (Central)
- Middle East (Bahrain)
- Middle East (UAE)

If you deploy IAM Identity Center in an opt-in Region, then you must enable this Region in all the accounts for which you want to manage access to IAM Identity Center. All accounts need this configuration, whether or not you'll create resources in that Region. You can enable a Region for the current accounts in your organization and you must repeat this action when you add new accounts. 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.

> **Note**
>
> Your AWS member account must be opted into the same Region as the opt-in Region where your IAM Identity Center instance is located, so you can access the AWS member account from the AWS access portal.

Metadata stored in opt-in Regions

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

AWS Regions that are enabled by default

The following Regions are enabled by default and you can enable IAM Identity Center in these Regions.

- 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)
- Asia Pacific (Sydney)
- Canada (Central)
- Asia Pacific (Osaka)

Switching AWS Regions

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. For more information, see Considerations for choosing an AWS Region.

You can switch your IAM Identity Center Region only by deleting your current IAM Identity Center instance and creating an instance in another Region. If you already enabled an AWS managed application with your existing IAM Identity Center instance, disable the application before deleting IAM Identity Center. For instructions on disabling AWS managed applications, see Disabling an AWS managed application.

> **Note**
>
> If you are considering switching your IAM Identity Center Region to enable the deployment of an AWS managed application in another Region, consider replicating your IAM Identity Center instance to that Region instead. For more information, see *Using IAM Identity Center across multiple AWS Regions*.

Configuration considerations in the new Region

You must recreate users, groups, permission sets, applications, and assignments in the new IAM Identity Center 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. Switching to a different Region also changes the URL for the AWS access portal, which provides your users with single sign-on access to their AWS accounts and applications. You might also need to recreate some IAM Identity Center configuration through the Management Console of your new instance.

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.

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, 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 is 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*.

Using IAM Identity Center for user access to applications only

You can use IAM Identity Center for user access to applications such as Kiro, 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 IAM Identity Center identity source tutorials.

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 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.

IAM roles created by IAM Identity Center

When you assign a user to an AWS account 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. Replicating your IAM Identity Center instance to additional Regions doesn't affect existing IAM roles, and it doesn't create new IAM roles.

> **Note**
>
> Permissions sets are not used to grant permissions to applications.

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 are 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 is 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 do not 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 Organization 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 an organization instance of 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). For more information, see Creating and managing an AWS Organization in the *AWS Organizations User Guide*.

When signed in with administrative credentials from an AWS Organizations member account, you can enable an account instance of IAM Identity Center. Account instances have limited capabilities and are bound to a single AWS account.

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*.

- Organization instance (recommended)

  An instance of IAM Identity Center that you enable in the AWS Organizations management account. Organization instances support all features of IAM Identity Center. We recommend that you deploy an organization instance rather than account instances to minimize the number of management points.

- Account instance

  An instance of IAM Identity Center that is bound to a single AWS account, and that is visible only within the AWS account and AWS Region in which it is enabled. Use an account instance for simpler, single-account scenarios. You can enable an account instance from either of the following:

  - An AWS account that isn't managed by AWS Organizations
  - A member account in AWS Organizations

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. You can associate only one instance of IAM Identity Center with a standalone AWS account and use that instance for application assignments within that standalone AWS account.

Use the following table to compare the capabilities provided by the instance type:

Capability	Instance in the AWS Organizations management account (recommended)	Instance in a member account	Instance in a standalone AWS account
Manage users	✅Yes	✅Yes	✅Yes
AWS access portal for single-sign on access to your AWS managed applications	✅Yes	✅Yes	✅Yes
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
Encryption at rest using a customer-managed KMS key	✅Yes	❌No	❌No

Capability	Instance in the AWS Organizations management account (recommended)	Instance in a member account	Instance in a standalone AWS account
Replicating IAM Identity Center to additional Regions	✅Yes	⊗No	⊗No

For more information about AWS managed applications and IAM Identity Center, see AWS managed applications that you can use with IAM Identity Center.

Topics

- Organization instances of IAM Identity Center
- Account instances 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 are 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 before November 15, 2023, you have an organization instance of IAM Identity Center.

To enable an organization instance of IAM Identity Center, see To enable an instance of IAM Identity Center.

When to use an organization instance

An organization instance is the primary method of enabling IAM Identity Center and usually, 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, assigning access to customer managed applications, and multi-Region replication.
- **Reduction of 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.
- **Central control of the 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).

For instructions on enabling an organization instance of IAM Identity Center, see To enable an instance of IAM Identity Center.

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 or a standalone AWS account that isn't managed by AWS Organizations.

For instructions on enabling an account instance of IAM Identity Center, see To enable an instance of IAM Identity Center and choose the **Account** tab.

When to use an account instance

In most cases, an organization instance is recommended. Use account instances 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.
- You do not control the AWS organization in which you operate. For example, a third-party controls the AWS organization that manages your AWS accounts.

> **Important**
>
> If you plan to use IAM Identity Center to support applications in multiple accounts, use an organization instance. Account instances do not support this use case.

AWS managed applications that support account instances

See AWS managed applications that you can use with IAM Identity Center 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.

Availability constraints for member accounts

To deploy account instances of IAM Identity Center in AWS Organizations member accounts, one of the following conditions must be true:

- There is no organization instance of IAM Identity Center in your organization.
- There is an organization instance of IAM Identity Center in your organization and the instance administrator permits creation of 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 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 Permit account instance creation in member accounts.

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 do not 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 IAM Identity Center Region data storage and operations.

Account instance considerations

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

- Account instances do not support permission sets and therefore do not support access to AWS accounts.
- You can't convert or 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 is created.

Permit account instance creation in member accounts

If you enabled IAM Identity Center before November 15, 2023, you have an organization instance of IAM Identity Center with the ability for member accounts to create account instances 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.

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

> **Important**
>
> Enabling account instances of IAM Identity Center for member accounts is a one-time operation. This means that this operation cannot 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.

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**.

Use Service Control Policies to control account instance creation

The ability for member accounts to create account instances depends on when you enabled IAM Identity Center:

- **Before November 2023** – You must permit account instance creation in member accounts, which is an action that cannot be reversed.
- **After November 15, 2023** – Member accounts can create account instances by default.

In either case, you can use Service Control Policies (SCPs) to:

- Prevent all member accounts from creating account instances.
- Allow only specific member accounts to create account instances.

Prevent account instances

Use the following procedure to generate an SCP that prevents member accounts from creating account instances of IAM Identity Center.

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. SCPs 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.*

Limit account instances

Instead of preventing all account instance creation, this policy denies any attempt to create an account instance of IAM Identity Center for all AWS accounts except those explicitly listed in the *"<ALLOWED-ACCOUNT-ID>"* placeholder.

Example: Deny policy to limit account instance creation

JSON

```
{

    "Version":"2012-10-17",
    "Statement" : [
        {
            "Sid": "DenyMemberAccountInstances",
            "Effect": "Deny",
            "Action": "sso:CreateInstance",
            "Resource": "*",
            "Condition": {
                 "StringNotEquals": {
                    "aws:PrincipalAccount": ["<ALLOWED-ACCOUNT-ID>"]
                }
            }
        }
    ]
}
```

- Replace [*"<ALLOWED-ACCOUNT-ID>"*] with the actual AWS account ID(s) that you want to allow to create an account instance of IAM Identity Center.
- You can list multiple allowed account IDs in the array format: [*"111122223333", "444455556666"*].
- Attach this policy to your organization SCP to enforce centralized control over IAM Identity Center account instance creation.

For instructions on attaching an SCP, see Attaching and detaching service control policies in the *AWS Organizations User Guide.*

Delete your IAM Identity Center instance

When an IAM Identity Center instance is deleted, all the data in that instance is deleted and cannot be recovered. The following table describes what data is deleted based on the directory type that is 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	Yes
All applications you have configured in IAM Identity Center	Yes	Yes
All user assignments you have configured for AWS accounts and applications	Yes	Yes
All users and groups in the directory or store	**N/A**	Yes

If you replicated your IAM Identity Center instance to additional Regions, you must remove those Regions before deleting the instance.

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 checkbox to acknowledge you understand that your data will be deleted. Type your IAM Identity Center instance in the text box, and then choose **Confirm**.

Enable IAM Identity Center

When you enable IAM Identity Center you choose an AWS IAM Identity Center instance type to enable. An instance of a service is a single deployment of a service within your AWS environment. There are two types of instances available for IAM Identity Center: organization instances and account instances. The instance types available for you to enable depend upon the account type you are signed into.

The following list identifies the type of IAM Identity Center instances you can enable for each type of AWS account:

- **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. You can replicate this instance type to additional Regions for enhanced resiliency of account access and flexibility in the choice of AWS application deployment Regions.
- **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. You can associate only one instance of IAM Identity Center with a standalone AWS account and use that instance for application assignments within that standalone AWS account.

> **Important**
>
> The organization management account can control whether organization member accounts can create account instances of IAM Identity Center by using a Service Control Policy.
> If you use a free tier account, creating an AWS organization automatically upgrades your account to a paid plan with pay-as-you-go pricing. Your free tier credits expire immediately. For more information, see AWS Free Tier FAQs.

For a comparison of the different capabilities provided by the different instance types, see Organization and account instances of IAM Identity Center.

Before enabling IAM Identity Center, we recommend you review the IAM Identity Center prerequisites and considerations.

To enable an instance of IAM Identity Center

Choose the tab for the type of IAM Identity Center instance you want to enable, either an organization or account instance:

Organization (recommended)

1. Do one 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 with a standalone AWS account (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
   - **Already using AWS Organizations (IAM credentials)** – Sign in using your management account credentials.
2. Open the IAM Identity Center console.
3. (Optional) If you want to use a customer managed KMS key for encryption at rest rather than the default AWS managed key, configure the customer managed key in the **Key for encrypting IAM Identity Center data at rest** section. For more information, refer to Implementing customer managed KMS keys in AWS IAM Identity Center.

   > **Important**
   >
   > Perform this step only if you've configured the necessary permissions for use of the KMS customer managed key. Without proper permissions, this step may fail or disrupt IAM Identity Center administration and AWS managed applications.

4. Under **Enable IAM Identity Center**, choose **Enable**.
5. On the **Enable IAM Identity Center with AWS Organizations** page, review the information and then select **Enable** to complete the process.

> **Note**
>
> AWS Organizations can have IAM Identity Center enabled only in a single AWS Region. After enabling IAM Identity Center, if you need to change the Region that IAM Identity Center is enabled in, you must delete the current instance and create an instance in the other Region.

After enabling your organization instance we recommend that you do the following steps to finish setting up your environment:

- Confirm that you are using the identity source of your choice. If you already have an assigned identity source, you can continue to use it. For more information, see Confirm your identity sources in IAM Identity Center.
- Register a member account as a delegated administrator. For more information, see Delegated administration.
- IAM Identity Center provides you an access portal to AWS resources. 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), see Update firewalls and gateways to allow access to the AWS access portal.

Account

1. Do one 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.
   - **Already using AWS Organizations (IAM credentials)** – Sign in using your member account administrative credentials.
2. Open the IAM Identity Center console.
3. If you are new to AWS or have a standalone AWS account, under **Enable IAM Identity Center**, choose **Enable**.

You see the **Enable IAM Identity Center with AWS Organizations** page. We recommend this option, but it is not required.

Select the link **enable an account instance of IAM Identity Center**.

4. If you are an administrator of an AWS Organizations member account, under **Enable an account instance of IAM Identity Center**, select **Enable an account instance**.
5. On the **Enable an account instance of IAM Identity Center** page, review the information and *optionally* add tags that you want to associate with this account instance. Then select **Enable** to complete the process.

> **Note**
>
> If your AWS account is a member of an organization, there might be restrictions on your ability to enable an account instance of IAM Identity Center.
>
> - If your organization enabled IAM Identity Center before November 15, 2023 the ability for member accounts to create account instances is disabled by default and must be enabled by the management account of the organization.
> - If your organization enabled IAM Identity Center after November 15, 2023 the ability for member account to create account instances is enabled by default. However, service control policies can be used to prevent the creation of account instances of IAM Identity Center within an organization.
>
> For more information, see the section called "Permit account instance creation in member accounts" and the section called "SCPs for account instance creation".

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 are using the identity source of your choice. If you already have an assigned identity source, you can continue to use it.

If you are 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 are 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.

To 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 is 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 are already managing users and groups in either your AWS Managed Microsoft AD directory using 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 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 IAM Identity Center identity source tutorials.

> **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.

> **Note**
>
> If you plan to replicate IAM Identity Center to additional Regions, you will need to configure an external identity provider. For more details including the prerequisites, see *Using IAM Identity Center across multiple AWS Regions*.

Update firewalls and gateways to allow access to the AWS access portal

The AWS access portal provides 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.

> **Note**
>
> AWS managed applications integrate with IAM Identity Center and use it for authentication and directory services, but might not use the AWS access portal for application access.

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 allowlist the domains and URL endpoints associated with the AWS access portal.

The following list provides the IPv4 and dual-stack domains and URL endpoints to add to your web-content filtering solution allowlists.

IPv4 allow list

- *[Directory ID or alias]*.awsapps.com
- *[Identity Center instance ID]*.*[Region]*.portal.amazonaws.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]*.prod.pr.panorama.console.api.aws/panoramaroute
- *[Region]*.signin.aws
- *[Region]*.signin.aws.amazon.com

- signin.aws.amazon.com
- *.cloudfront.net
- opfcaptcha-prod.s3.amazonaws.com

Dual-stack allow list

- *[Identity Center instance ID]*.portal.*[Region]*.app.aws
- *.aws.dev
- *.awsstatic.com
- *.console.aws.a2z.com
- oidc.*[Region]*.api.aws
- sso.*[Region]*.api.aws
- portal.sso.*[Region]*.api.aws
- *[Region]*.sso.signin.aws
- *[Region]*.signin.aws.amazon.com
- signin.aws.amazon.com
- *.cloudfront.net
- cdn.us-east-1.threat-mitigation.aws.amazon.com
- us-east-1.threat-mitigation.aws.amazon.com
- amcs-captcha-prod-us-east-1.s3.dualstack.us-east-1.amazonaws.com

Combined allow list (IPv4 + Dual-stack with backward compatibility)

- *[Directory ID or alias]*.awsapps.com
- *[Identity Center instance ID]*.*[Region]*.portal.amazonaws.com
- *[Identity Centers instance ID]*.portal.*[Region]*.app.aws
- *.aws.dev
- *.awsstatic.com
- *.console.aws.a2z.com
- oidc.*[Region]*.amazonaws.com
- oidc.*[Region]*.api.aws
- *.sso.amazonaws.com

- `*.sso.[Region].amazonaws.com`
- `sso.[Region].api.aws`
- `*.sso-portal.[Region].amazonaws.com`
- `portal.sso.[Region].api.aws`
- `[Region].prod.pr.panorama.console.api.aws/panoramaroute`
- `[Region].signin.aws`
- `[Region].sso.signin.aws`
- `[Region].signin.aws.amazon.com`
- `signin.aws.amazon.com`
- `*.cloudfront.net`
- `opfcaptcha-prod.s3.amazonaws.com`
- `cdn.us-east-1.threat-mitigation.aws.amazon.com`
- `us-east-1.threat-mitigation.aws.amazon.com`
- `amcs-captcha-prod-us-east-1.s3.dualstack.us-east-1.amazonaws.com`

Considerations for allowlisting domains and URL endpoints

In addition to the allowlist requirements for the AWS access portal, the other services and applications you use might require allowlisting of domains.

- 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.
- 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.

IAM Identity Center identity source tutorials

You can connect your existing identity source in your AWS Organizations management account to an organization instance of IAM Identity Center. If you do not 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 are 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 Enable 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
- Video tutorials

Using Active Directory as an identity source

If you are managing users in either your AWS Managed Microsoft AD directory using 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 is 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 are 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 are already using Active Directory , the following topics will help you prepare to connect your directory to IAM Identity Center.

> **Note**
>
> If you plan to connect an AWS Managed Microsoft AD directory or a self-managed directory in Active Directory and you are not using RADIUS MFA with AWS Directory Service, enable MFA in IAM Identity Center.

AWS Managed Microsoft AD

1. Review the guidance in 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 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.

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. For more information, see Using SAML and SCIM identity federation with external identity providers.

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 the SCIM endpoint and access token. You'll 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 your IdP 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.

> **Note**
>
> Google Workspace doesn't currently support SAML multiple assertion consume service (ACS) URLs in the AWS IAM Identity Center application. This SAML feature is required to fully take advantage of multi-Region support in IAM Identity Center. If you plan to replicate IAM Identity Center to additional Regions, be aware that using a single ACS URL may affect the user experience in those additional Regions. Your primary Region will continue to function normally. We recommend that you work with your IdP vendor to enable this feature. For more information about the user experience in additional Regions with a single ACS URL, see the section called "Using AWS managed applications without multiple ACS URLs" and the section called "AWS account access resiliency without multiple ACS URLs".

User information from Google Workspace is synchronized into IAM Identity Center using the System for Cross-domain Identity Management (SCIM) 2.0 protocol. For more information, see Using SAML and SCIM identity federation with external identity providers.

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 identity provider and connect with your IAM Identity Center.

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 Enable 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 *Enable 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,

      - IPv4`https://scim.`*`Region`*`.amazonaws.com/`*`11111111111-2222-3333-4444-555555555555`*`/scim/v2`
      - Dual-stack`https://scim.`*`Region`*`.api.aws/`*`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.

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.

Passing attributes for access control - *Optional*

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.

Assign access to AWS accounts

The following steps are only required to grant access to AWS accounts only. These steps are not required to grant access to AWS applications.

Note

To complete this step, you'll need an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

Step 1: 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 cannot 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 2: 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.

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.

Troubleshooting

For general SCIM and SAML troubleshooting with Google Workspace, see the following sections:

- Specific users fail to synchronize into IAM Identity Center from an external SCIM provider
- Issues regarding contents of SAML assertions created by IAM Identity Center
- Duplicate user or group error when provisioning users or groups with an external identity provider
- For Google Workspace troubleshooting, see Google Workspace documentation.

The following resources can help you troubleshoot as you work with AWS:

- AWS re:Post - Find FAQs and links to other resources to help you troubleshoot issues.
- AWS Support - Get technical support

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. For more information, see Using SAML and SCIM identity federation with external identity providers.

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 .
- If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts from those Regions. For more details, see the section called “Step 3: Update external IdP setup”. See the JumpCloud documentation for additional details.
- 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 the SCIM endpoint and access token. You'll 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 your IdP 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. For more information, see Using SAML and SCIM identity federation with external identity providers.

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 cannot 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 configurable 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 is 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**.
5.

Example

<caption>On the **Select permission sets** page, choose the `RegionalAdmin` permission set created in Step 2.1, and then choose **Next**.</caption>

6. On the **Review and submit** page, review your selections and then choose **Submit**.

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 **Default IPv4** or **Dual-stack**. You can download the service provider metadata file after you complete the identity source change.
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 (Assertion Consumer Service URL)** values now point to endpoints in AWS.

   - **Identifier** - This is the **Issuer URL** from IAM Identity Center. The same value applies regardless of whether you use IPv4-only or dual-stack endpoints.
   - **Reply URL (Assertion Consumer Service URL)** - The values here include both IPv4-only and dual-stack endpoints from all enabled Regions of your IAM Identity Center. You can use the primary Region's ACS URL as the default one so that users are redirected to the primary Region when they launch the Amazon Web Services application from Microsoft Entra ID. For more information on ACS URLs, see ACS endpoints in the primary and additional AWS Regions.
   - (Optional) If you replicated IAM Identity Center to additional Regions, you can also create a bookmark app in Microsoft Entra ID for the AWS access portal in each additional Region. This enables your users to access the AWS access portal in additional Regions from Microsoft Entra ID. Make sure to grant your users permissions to access the bookmark apps in Microsoft Entra ID. See Microsoft Entra ID documentation for more details. If you plan to replicate IAM Identity Center to additional Regions later, visit Microsoft Entra ID configuration for access to additional Regions for guidance how to enable access to the additional Regions after this initial setup.
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 do not 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 is 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,
      - IPv4: `https://scim.`*`aws-region`*`.amazonaws.com/`*`11111111111-2222-3333-4444-555555555555`*`/scim/v2`
      - Dual stack: `https://scim.`*`aws-region`*`.api.aws/`*`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.

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 4.2`**. 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.

Step 5: Configure ABAC - *Optional*

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.

Assign access to AWS accounts

The following steps are only required to grant access to AWS accounts only. These steps are not required to grant access to AWS applications.

> **Note**
>
> To complete this step, you'll need an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

Step 1: IAM Identity Center: Grant Microsoft Entra ID 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.

Step 2: Microsoft Entra ID: Confirm Microsoft Entra ID users access to AWS resources

1. Return to the **Microsoft Entra ID** console and navigate to your IAM Identity Center SAML-based Sign-on application.
2. Select **Users and groups** and select **Add users or groups**. You'll add the user you created in this tutorial in Step 4 to the Microsoft Entra ID application. By adding the user, you'll allow them to sign-in to AWS. Search for the user you created at Step 4. If you followed this step, it would be **`RichardRoe`**.
   - For a demo, see Federate your existing IAM Identity Center instance with Microsoft Entra ID

Microsoft Entra ID configuration for access to additional Regions of IAM Identity Center - Optional

If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts through the additional Regions. The steps below guide you through the procedure. For more details about this topic including the prerequisites, see *Using IAM Identity Center across multiple AWS Regions*.

1. Retrieve ACS URLs for the additional regions from the IAM Identity Center console as outlined in ACS endpoints in the primary and additional AWS Regions.
2. In the Microsoft Entra admin center console, navigate to **Identity > Applications > Enterprise applications** and then choose **AWS IAM Identity Center**.
3. On the left, choose **2. Set up Single sign-on**.
4. On the **Basic SAML Configuration** page, under **Reply URL (Assertion Consumer Service URL)** section, choose **Add reply URL** for each additiona Region's ACS URL. You can keep the primary

Region's ACS URL as the default ones so that users keep getting redirected to the primary Region when they launch the AWS IAM Identity Center application from Microsoft Entra ID.

5. When done adding the ACS URLs, save the **AWS IAM Identity Center** application.
6. You can create a bookmark app in Microsoft Entra ID for the AWS access portal in each additional Region. This enables your users to access the AWS access portal in additional Regions from Microsoft Entra ID. Make sure to grant your users permissions to access the bookmark apps in Microsoft Entra ID. See Microsoft Entra ID documentation for more details.
7. Verify that you can sign into the AWS access portal in each additional Region. Navigate to the AWS access portal URLs or launch the bookmark apps from Microsoft Entra ID.

Troubleshooting

For general SCIM and SAML troubleshooting with Microsoft Entra ID, see the following sections:

- Synchronization issues with Microsoft Entra ID and IAM Identity Center
- Specific users fail to synchronize into IAM Identity Center from an external SCIM provider
- Issues regarding contents of SAML assertions created by IAM Identity Center
- Duplicate user or group error when provisioning users or groups with an external identity provider
- Additional resources

Synchronization issues with Microsoft Entra ID and IAM Identity Center

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 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.

Microsoft Entra ID Guest Account Synchronization

If you would like to sync your Microsoft Entra ID guest users to IAM Identity Center, see the following procedure.

Microsoft Entra ID guest users' email is different than Microsoft Entra ID users. This difference causes issues when attempting to synchronize Microsoft Entra ID guest users with IAM Identity Center. For example, see the following email address for a guest user:

`exampleuser_domain.com#EXT#@domain.onmicrosoft.com.`

IAM Identity Center does not expect the email address to contain the *`#EXT#@domain`* format.

1. Sign in to the Microsoft Entra admin center and navigate to **Identity** > **Applications** > **Enterprise applications** and then choose **AWS IAM Identity Center**
2. Navigate to the **Single Sign On** tab in the left pane.
3. Select **Edit** which appears next to **User Attributes & Claims**.
4. Select **Unique User Identifier (Name ID)** following **Required Claims**.
5. You will create two claim conditions for your Microsoft Entra ID users and guest users:
   a. For Microsoft Entra ID users, create a user type for members with source attribute set to `user.userprincipalname`.
   b. For Microsoft Entra ID guest users, create a user type for external guests with the source attribute set to `user.mail`.

c. Select **Save** and retry signing in as a Microsoft Entra ID guest user.

Additional resources

- For general SCIM troubleshooting tips, see Troubleshooting IAM Identity Center issues.
- For Microsoft Entra ID troubleshooting, see Microsoft documentation.
- To learn more about federation across multiple AWS accounts, see Securing AWS accounts with Azure Active Directory Federation.

The following resources can help you troubleshoot as you work with AWS:

- AWS re:Post - Find FAQs and links to other resources to help you troubleshoot issues.
- AWS Support - Get technical support

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. For more information, see Using SAML and SCIM identity federation with external identity providers.

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.

- Import users – Users can be imported from IAM Identity Center to Okta.

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.

The following features are supported when connecting Okta to IAM Identity Center through SAML:

- IdP-initiated SAML sign-in – Users sign in through the Okta portal and gain access to IAM Identity Center.
- SP-initiated SAML sign-in – Users access the AWS access portal, which redirects them to sign in through the Okta portal.

> **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 Enable 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 cannot 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.
- If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts through additional Regions. For more details including the prerequisites, see *Using IAM Identity Center across multiple AWS Regions*. Okta-specific steps are described in Okta configuration for access to additional Regions

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**, copy the following items to a text file for easy access:

      - **IAM Identity Center Assertion Consumer Service (ACS) URL** – You have a choice between IPv4-only and dual-stack ACS URLs. Also, if your IAM Identity Center instance is enabled in multiple Regions, each additional Region has its own IPv4-only and dual-stack ACS URLs. For more information on ACS URLs, see ACS endpoints in the primary and additional AWS Regions.
      - **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(s) you copied for **IAM Identity Center Assertion Consumer Service (ACS) URL**. You can use the primary Region's ACS URL as the default one so that users are redirected to the primary Region when they launch the Amazon Web Services application from Okta.
- (Optional) If you replicated IAM Identity Center to additional Regions, you can also create a bookmark app in Okta for the AWS access portal in each additional Region. This enables your users to access the AWS access portal in additional Regions from Okta. Make sure to grant your users permissions to access the bookmark apps in Okta. See Okta documentation for more details. If you plan to replicate IAM Identity Center to additional Regions later, visit Okta configuration for access to additional Regions for guidance how to enable access to the additional Regions after this initial setup.
- 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** - The endpoint format depends on your configuration:

      - IPv4: https://scim.*us-east-2*.amazonaws.com/*11111111111-2222-3333-4444-555555555555*/scim/v2
      - Dual-stack: https://scim.*us-east-2*.api.aws/*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.

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.

Passing attributes for access control - *Optional*

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.

Assign access to AWS accounts

The following steps are only required to grant access to AWS accounts only. These steps are not required to grant access to AWS applications.

> **Note**
>
> To complete this step, you'll need an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

Step 1: 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 2: 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.

You can also use the AWS access portal. This redirects you to sign in through the Okta portal before taking you to the AWS access portal. This path follows the SP-initiated SAML sign-in flow.

Okta configuration for access to additional Regions of IAM Identity Center - Optional

If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts through the additional Regions. The steps below guide you through the procedure. For more details about this topic including the prerequisites, see *Using IAM Identity Center across multiple AWS Regions*.

1. Retrieve ACS URLs for the additional regions from the IAM Identity Center console as outlined in ACS endpoints in the primary and additional AWS Regions.
2. In the Okta admin dashboard's navigation pane, choose **Applications**, and then **Applications** again in the expanded list.
3. Choose the **AWS IAM Identity Center** application.
4. Choose the **Sign On** tab.
5. Under **Advanced Sign-on Settings**, and **Other Requestable SSO URLs**, choose **Add another** for each additional Region's ACS URL, and paste the ACS URL into the text field.
6. When done adding the ACS URLs, save the **AWS IAM Identity Center** application.
7. You can create a bookmark app in Okta for the AWS access portal in each additional Region. This enables your users to access the AWS access portal in additional Regions from Okta. Make sure to grant your users permissions to access the bookmark apps in Okta. See Okta documentation for more details.
8. Verify that you can sign into the AWS access portal in each additional Region. Navigate to the AWS access portal URLs or launch the bookmark apps from Okta.

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 or group 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.

Troubleshooting

For general SCIM and SAML troubleshooting with Okta, see the following sections:

- Reprovisioning users and groups deleted from IAM Identity Center
- Automatic Provisioning Error in Okta
- Specific users fail to synchronize into IAM Identity Center from an external SCIM provider
- Issues regarding contents of SAML assertions created by IAM Identity Center
- Duplicate user or group error when provisioning users or groups with an external identity provider
- Additional resources

Reprovisioning users and groups deleted from IAM Identity Center

- You could receive the following error message in the Okta Console, if you're attempting to change either a user or group in Okta that was once synchronized and then deleted from IAM Identity Center:
  - Automatic profile push of user `Jane Doe` to app AWS IAM Identity Center failed: Error while trying to push profile update for `jane_doe@example.com`: No user returned for user `xxxxx-xxxxxx-xxxxx-xxxxxxx`
  - Linked group is missing in AWS IAM Identity Center. Change the linked group to resume pushing group memberships.
- You could also receive the following error message in the Okta's Systems Logs for either synchronized and deleted IAM Identity Center users or groups:
  - Okta Error: Eventfailed application.provision.user.push_profile : No user returned for user `xxxxx-xxxxxx-xxxxx-xxxxxxx`
  - Okta Error: application.provision.group_push.mapping.update.or.delete.failed.with.error : Linked group is missing in AWS IAM Identity Center. Change the linked group to resume pushing group memberships.

> **Warning**
>
> Users and groups should be deleted from Okta rather than IAM Identity Center if you have synchronized Okta and IAM Identity Center using SCIM.

Troubleshooting deleted IAM Identity Center Users

To address this issue with deleted IAM Identity Center users, the users must be deleted from Okta. If necessary, these users would also need to be recreated in Okta. When the user is recreated in Okta, it will also be reprovisioned into the IAM Identity Center through SCIM. For more information on deleting a user, see Okta documentation.

> **Note**
>
> If you need to remove a Okta user's access to IAM Identity Center, you should first remove them from their Group Push and then their Assignment Group in Okta. This ensures the user is removed from their associated group membership in IAM Identity Center. For more information on troubleshooting Group Push, see Okta documentation.

Troubleshooting deleted IAM Identity Center Groups

To address this issue with deleted IAM Identity Center groups, the group must be deleted from Okta. If necessary, these groups would also need to be recreated in Okta using Group Push. When the user is recreated in Okta, it will also be reprovisioned into the IAM Identity Center through SCIM. For more information on deleting a group, see Okta documentation.

Automatic Provisioning Error in Okta

If you receive the following error message in Okta:

Automatic provisioning of user Jane Doe to app AWS IAM Identity Center failed: Matching user not found

See Okta documentation for more information.

Additional resources

- For general SCIM troubleshooting tips, see Troubleshooting IAM Identity Center issues.

The following resources can help you troubleshoot as you work with AWS:

- AWS re:Post - Find FAQs and links to other resources to help you troubleshoot issues.
- AWS Support - Get technical support

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. For more information, see Using SAML and SCIM identity federation with external identity providers.

> **Note**
>
> OneLogin doesn't currently support SAML multiple assertion consume service (ACS) URLs in the AWS IAM Identity Center application. This SAML feature is required to fully take advantage of multi-Region support in IAM Identity Center. If you plan to replicate IAM Identity Center to additional Regions, be aware that using a single ACS URL may affect the user experience in those additional Regions. Your primary Region will continue to function normally. We recommend that you work with your IdP vendor to enable this feature. For more information about the user experience in additional Regions with a single ACS URL, see the section called "Using AWS managed applications without multiple ACS URLs" and the section called "AWS account access resiliency without multiple ACS URLs".

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 the SCIM endpoint and access token. You'll 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 your IdP 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 navigation pane.

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 navigation pane, choose **Provisioning**.
6. Select the check boxes for **Enable provisioning**, **Create user**, **Delete user**, and **Update user**, and then choose **Save**.
7. In the navigation pane, 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 navigation pane.
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. For more information, see Using SAML and SCIM identity federation with external identity providers.

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
- 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
- Troubleshooting

Prerequisites

You'll 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.
- If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts from those Regions. For more details, see the section called “Step 3: Update external IdP setup”. See the PingFederate documentation for additional details.

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 the SCIM endpoint and access token. You'll 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 your IdP 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 **`AWSIAMIdentityCenterchannel`**), 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.

Troubleshooting

For general SCIM and SAML troubleshooting with PingFederate, see the following sections:

- Specific users fail to synchronize into IAM Identity Center from an external SCIM provider
- Issues regarding contents of SAML assertions created by IAM Identity Center
- Duplicate user or group error when provisioning users or groups with an external identity provider
- For more information on PingFederate, see PingFederate documentation.

The following resources can help you troubleshoot as you work with AWS:

- AWS re:Post - Find FAQs and links to other resources to help you troubleshoot issues.
- AWS Support - Get technical support

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.

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
- 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
- Troubleshooting

Prerequisites

You'll 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.

- If you replicated IAM Identity Center to additional Regions, you must update your identity provider configuration to enable access to AWS managed applications and AWS accounts from those Regions. For more details, see the section called "Step 3: Update external IdP setup". See the PingOne documentation for additional details.

Considerations

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

- 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 Provision users and groups from an external identity provider 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 the SCIM endpoint and access token. You'll 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 your IdP 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 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.

Troubleshooting

For general SCIM and SAML troubleshooting with PingOne, see the following sections:

- Specific users fail to synchronize into IAM Identity Center from an external SCIM provider
- Issues regarding contents of SAML assertions created by IAM Identity Center
- Duplicate user or group error when provisioning users or groups with an external identity provider
- For more information on PingOne, see PingOne documentation.

The following resources can help you troubleshoot as you work with AWS:

- AWS re:Post - Find FAQs and links to other resources to help you troubleshoot issues.
- AWS Support - Get technical support

Configure user access with the default IAM Identity Center directory

When you enable IAM Identity Center for the first time, it is automatically configured with an Identity Center directory as your default identity source, so you do not need to choose an identity source. If your organization uses another identity provider such as 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'll use the default directory as your identity source and an IAM Identity Center organization instance to set up and test an administrative user. This administrative user creates and manages users and groups and grants AWS access with permission sets. In the next steps, you'll 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'll 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. Alternatively, if you want to grant users access to application, you can follow step 1 of this procedure and configure application access.

Prerequisites

The following prerequisites are needed to complete this tutorial:

- Enable IAM Identity Center and have an organization instance of IAM Identity Center.
  - If you have an account instance of IAM Identity Center, you can create users and groups as well as grant them access to applications. For more information, see Application access.
- Sign in to the AWS Management Console and access the IAM Identity Center console either as a:
  - **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.
    - For more help signing in to the AWS Management Console, see AWS Sign-In Guide.
- You can configure multi-factor authentication for your IAM Identity Center users. For more information, see Configure MFA in IAM Identity Center.

Step 1: Add a user

1. Open the IAM Identity Center console.
2. In the IAM Identity Center navigation pane, choose **Users**, then select **Add user**.
3. 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 cannot 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**. 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.

4. 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
5. 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**.
6. 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'll add administrative permissions for the *Admin team* group so that *Nikki* has access to resources.

Step 2: Add administrative permissions

> **Important**
>
> Follow these steps only if you enabled an organization instance of IAM Identity Center.

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 **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.

> **Congratulations!**
>
> You have successfully set up your first user, group, and permission set.

In the next portion of this tutorial you'll 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 are granted permission according to their permission set. To verify that the user is correctly configured, in this next step you'll 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'll 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 and register their MFA device.

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 are assigned.

Video tutorials

As an additional resource, you can use these video tutorials to learn more about setting up external identity providers:

- Migrating between external identity providers in AWS IAM Identity Center
- Federating your existing AWS IAM Identity Center instance with Microsoft Entra ID

Set up your workforce in IAM Identity Center

IAM Identity Center is the AWS solution for connecting your workforce users to AWS managed applications such as Kiro and Amazon Quick, 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 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.

Choose the approach that best fits your organization's identity management strategy and existing infrastructure.

Topics

- Users, groups, and provisioning in IAM Identity Center
- Manage your identity source
- Manage users in the Identity Center directory

Users, groups, and provisioning in IAM Identity Center

IAM Identity Center enables you to control who can sign in and what resources they can access. A user must be provisioned to sign in. You can then assign access only to provisioned users or groups. Learn about users, groups, and provisioning in IAM Identity Center.

Username and email address uniqueness

IAM Identity Center requires each user have a unique username. The username is the user's primary identifier. The username does not have to match the user's email address. IAM Identity Center requires that all usernames 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 doesn't support nested groups (A group within a group). 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 connect your identity source to IAM Identity Center. With IAM Identity Center, you are able to assign users and groups access to connected applications and AWS accounts.

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

User and group deprovisioning

Deprovisioning is the process of removing users and group information from IAM Identity Center.

If you're using Active Directory or an external identity provider with IAM Identity Center, you should remove users and groups from these identity sources rather than IAM Identity Center. Deleting IAM Identity Center users and groups will not completely remove them if your identity source is Active Directory or an external identity provider.

If you need to deprovision IAM Identity Center users or groups, you should first remove any assignments of permission sets or applications to the users or groups you want to deprovision. Otherwise, you'll have unassigned permission sets and application assignments in your IAM Identity Center.

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.

- **Your on premises or AWS managed Active Directory** – Choose this option if you want to connect your Active Directory (AD).
- **Identity Center directory** – When you enable IAM Identity Center for the first time, it is 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
- Supported user and group attributes in IAM Identity Center
- External identity providers
- Microsoft AD directory

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 are 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 directory and Active Directory

If you are 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.

> **Important**
>
> When changing your identity source type in IAM Identity Center to or from Active Directory be aware that the Identity Store ID will change. This can have the following implications:
>
> - Your default AWS access portal URL will change. You will need to communicate the new URL to your workforce and update bookmarks, gatewall or firewall allow-lists, and configurations where this URL is referenced. We recommend you perform this change in a scheduled maintenance window to minimize disruption to your users.
> - If you are using a customer managed KMS key for encryption at rest in IAM Identity Center, and have configured the KMS key policy with the encryption context, be aware that the encryption context for the Identity Store will change. For instance, in the Identity Store ARN "arn:aws:identitystore::123456789012:identitystore/d-922763e9b3", "d-922763e9b3" is the Identity Store ID. To prevent service disruption during this transition, temporarily modify your KMS key policy to use a wildcard pattern: "arn:aws:identitystore::123456789012:identitystore/*".

If you are 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.
- If users are deleted or disabled in the IAM Identity Center console using Identity Store APIs, users with active sessions can continue to access integrated applications and accounts. For information about authentication session duration and user behavior, see Understanding authentication sessions in IAM Identity Center.

For information about how IAM Identity Center provisions users and groups, see 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 will not 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 the section called "End active sessions for workforce users".

If users are deleted or disabled in the IAM Identity Center console using Identity Store APIs, users with active sessions can continue to access integrated applications and accounts. For information about authentication session duration and user behavior, see Understanding authentication sessions in IAM Identity Center.

> **Note**
>
> You cannot 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 External identity providers.

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 the fourth hour, active user sessions continue to run for an additional four hours. To revoke user sessions, see the section called "End active sessions for workforce users".

  If users are deleted or disabled in the IAM Identity Center console using Identity Store APIs, users with active sessions can continue to access integrated applications and accounts. For information about authentication session duration and user behavior, see Understanding authentication sessions in IAM Identity Center.

> **Note**
>
> You will not be able to revoke user sessions from the IAM Identity Center console after you've deleted the user.

- **Multi-Region support** – If you have replicated IAM Identity Center to additional Regions or plan to do so, you must use an external identity provider as the identity source. For more information including other prerequisites, see *Using IAM Identity Center across multiple AWS Regions*.

For information about how IAM Identity Center provisions users and groups, see Manage users in the Identity Center directory.

Changing from one external IdP to another external IdP

If you are 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 the fourth hour, active user sessions persist for an additional four hours. To revoke user sessions, see the section called “End active sessions for workforce users”.

  If users are deleted or disabled in the IAM Identity Center console using Identity Store APIs, users with active sessions can continue to access integrated applications and accounts. For information

about authentication session duration and user behavior, see Understanding authentication sessions in IAM Identity Center.

> **Note**
>
> You cannot 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 External identity providers.

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.
- If users are deleted or disabled in the IAM Identity Center console using Identity Store APIs, users with active sessions can continue to access integrated applications and accounts. For information about authentication session duration and user behavior, see Understanding authentication sessions in IAM Identity Center.
- **Multi-Region support** – If you have replicated IAM Identity Center to additional Regions or plan to do so, you must use an external identity provider as the identity source. For more information including other prerequisites, see *Using IAM Identity Center across multiple AWS Regions*.

For information about how IAM Identity Center provisions users and groups, see 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. To complete this procedure, you'll need an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

> **Warning**
>
> Depending on your current deployment, this change removes any user and group assignments that you configured in IAM Identity Center. This change will also remove permission set IAM roles from your AWS accounts. As a result, you may need to update your resource policies, and should ensure this will not disrupt your access to AWS KMS keys and Amazon EKS clusters. To learn more, see Referencing permission sets in resource policies, Amazon EKS Cluster config maps, and AWS KMS key policies.
> When this occurs, all users and groups, 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.

> **Note**
>
> If you replicated IAM Identity Center to additional Regions, you won't be able to change your identity source type. You can only replace the current external IdP with another one. To change the identity source type, you will need to remove all additional Regions first. For more information, see *Using IAM Identity Center across multiple AWS Regions*

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.

Supported user and group attributes in IAM Identity Center

This guide provides a reference for SCIM attribute support in IAM Identity Center. It lists which user and group attributes from the SCIM specification are supported in the IAM Identity Center identity store, and identifies specific attributes and sub-attributes that aren't supported.

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 through both manual entry and automatic SCIM provisioning.

- For information about the System for Cross-Domain Identity Management (SCIM) specification, see https://tools.ietf.org/html/rfc7642.
- For information about manual and automatic provisioning, see Provisioning when users come from an external IdP.
- For information about attribute mapping, see Attribute mappings between IAM Identity Center and External Identity Providers directory.

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 aren't supported by IAM Identity Center.

User objects not supported

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 not supported

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).

External identity providers

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 are using a self-managed directory in Active Directory or an AWS Managed Microsoft AD, see 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.

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 Provision users and groups from an external identity provider 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.

Topics

- 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

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 is 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.

   > **Note**
   >
   > The SAML metadata file you download contains both IPv4-only and dual-stack assertion consumer service (ACS) URLs. Further, if your IAM Identity Center is replicated to additional Regions, the metadata file contains ACS URLs for each additional Region. If your external IdP has a limit on the number of ACS URLs, you will need to remove the unnecessary ACS URLs. For example, if your organization has fully adopted dual-stack endpoints and no longer uses IP4v-only ones, you can remove the latter. An alternative approach is to not use the metadata file but to copy and paste the ACS URLs into the external IdP.

   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 is stored in IAM Identity Center. To complete this procedure, you'll need an Organization instance of IAM Identity Center. For more information, see Organization and account instances of 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.
- IAM Identity Center does not support signing SAML authentication requests that it sends to external IdPs.
- The IdP has to support multiple assertion consumer service (ACS) URLs if you plan to replicate IAM Identity Center to additional Regions, and fully take advantage of the benefits of a multi-Region IAM Identity Center. For more information, see *Using IAM Identity Center across multiple AWS Regions*. Using a single ACS URL may affect the user experience in additional Regions. Your primary Region will continue to function normally. For more information about the user experience in additional Regions with a single ACS URL, see the section called "Using AWS managed applications without multiple ACS URLs" and the section called "AWS account access resiliency without multiple ACS URLs".

- **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 IAM Identity Center identity source tutorials, 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 Provision users and groups from an external identity provider using SCIM. For additional details about IAM Identity Center's SCIM implementation, see the IAM Identity Center SCIM Implementation Developer Guide.

Topics

- Provision users and groups from an external identity provider using SCIM
- Rotate SAML 2.0 certificates

Provision users and groups from an external identity provider 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
- Generate an access token
- Enable automatic provisioning
- Delete an access token
- Disable automatic provisioning
- Rotate an access token
- Audit and reconcile auto-provisioned resources
- 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 IAM Identity Center identity source tutorials 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, multivalue attributes (such as multiple emails or phone numbers for a given user) are not provisioned with SCIM. Attempts to synchronize multivalue 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 multivalue 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.

- After setting up automatic provisioning with SCIM for your IdP, you can no longer add or edit users in the IAM Identity Center console. If you need to add or modify a user, you must do so from your external IdP or identity source.

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.

Generate an access token

Use the following procedure to generate a new access token in the IAM Identity Center console.

> **Note**
>
> This procedure requires that you have previously enabled automatic provisioning. For more information, see 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**.

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 IAM Identity Center identity source tutorials 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 the SCIM endpoint and access token. You'll 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 your IdP 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 IAM Identity Center identity source tutorials for your IdP.

Delete an access token

Use the following procedure to delete an existing access token in the IAM Identity Center console.

To 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**.

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 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**.

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 Generate an 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 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.

Audit and reconcile auto-provisioned resources

SCIM enables you to automatically provision users, groups, and group memberships from your identity source to IAM Identity Center. This guide helps you verify and reconcile these resources to maintain accurate synchronization.

Why audit your resources?

Regular auditing helps ensure your access controls remain accurate and your identity provider (IdP) stays properly synchronized with IAM Identity Center. This is particularly important for security compliance and access management.

Resources you can audit:

- Users
- Groups
- Group memberships

You can use AWS Identity Store APIs or CLI commands to conduct the audit and reconciliation. The following examples use AWS CLI commands. For API alternatives, refer to the corresponding operations in the *Identity Store API reference*.

How to audit resources

Here are examples for how to audit these resources using AWS CLI commands.

Before you begin, ensure you have:

- Administrator access to IAM Identity Center.
- AWS CLI installed and configured. For information, see the *AWS Command Line Interface User Guide*.
- Required IAM permissions for Identity Store commands.

Step 1: List current resources

You can view your current resources using the AWS CLI.

> **Note**
>
> When using the AWS CLI, pagination is handled automatically unless you specify `--no-paginate`. If you're calling the API directly (for example, with an SDK or a custom script), handle the `NextToken` in the response. This ensures you retrieve all results across multiple pages.

Example for users

```
aws identitystore list-users \
  --region REGION \
  --identity-store-id IDENTITY_STORE_ID
```

Example for groups

```
aws identitystore list-groups \
  --region REGION \
  --identity-store-id IDENTITY_STORE_ID
```

Example for group memberships

```
aws identitystore list-group-memberships \
  --region REGION \
  --identity-store-id IDENTITY_STORE_ID
  --group-id GROUP_ID
```

Step 2: Compare with your identity source

Compare the listed resources with your identity source to identify any discrepancies, such as:

- Missing resources that should be provisioned in IAM Identity Center.
- Extra resources that should be removed from IAM Identity Center.

Example for users

```
# Create missing users
aws identitystore create-user \
  --identity-store-id IDENTITY_STORE_ID \
  --user-name USERNAME \
  --display-name DISPLAY_NAME \
  --name GivenName=FIRST_NAME,FamilyName=LAST_NAME \
  --emails Value=EMAIL,Primary=true

# Delete extra users
aws identitystore delete-user \
  --identity-store-id IDENTITY_STORE_ID \
  --user-id USER_ID
```

Example for groups

```
# Create missing groups
aws identitystore create-group \
  --identity-store-id IDENTITY_STORE_ID \
```

```
  [group attributes]

# Delete extra groups
aws identitystore delete-group \
  --identity-store-id IDENTITY_STORE_ID \
  --group-id GROUP_ID
```

Example for group memberships

```
# Add missing members
aws identitystore create-group-membership \
  --identity-store-id IDENTITY_STORE_ID \
  --group-id GROUP_ID \
  --member-id '{"UserId": "USER_ID"}'

# Remove extra members
aws identitystore delete-group-membership \
  --identity-store-id IDENTITY_STORE_ID \
  --membership-id MEMBERSHIP_ID
```

Considerations

- Commands are subject to service quotas and API throttling.
- When you find many differences during reconciliation, make small, gradual changes to AWS Identity Store. This helps you avoid mistakes that affect multiple users.
- SCIM synchronization can override your manual changes. Check your IdP settings to understand this behavior.

Manual provisioning

Some IdPs do not 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 Username 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.

Rotate 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

The **Manage SAML 2.0 certificates** page displays colored status indicator icons 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 expired.
- **Yellow** – Indicates that a certificate expires in 90 days or less.
- **Green** – Indicates that a certificate is valid and remains 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.

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 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 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 Directory Service resides within your AWS Organizations management account.
- IAM Identity Center does not support SAMBA 4-based Simple AD as a connected directory.
- IAM Identity Center cannot synchronize Foreign Security Principals (FSPs). If a group in AWS Managed Microsoft AD contains members from a trusted domain as FSPs, those members will not sync.

For a demonstration on the process of using Active Directory as an identity source for IAM Identity Center, see the following YouTube video:

Using Active Directory as an identity source for AWS IAM Identity Center | Amazon Web Services

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 are 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 are 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.

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

AWS Managed Microsoft AD

1. Review the guidance in 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 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 External identity providers.
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 Active 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.

Provisioning when users come from Active Directory

IAM Identity Center uses the connection provided by the 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 above 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 between IAM Identity Center and External Identity Providers directory
- IAM Identity Center configurable AD sync

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 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 Directory Service directory users from trusted domains, the 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 Directory Service for Microsoft Active Directory.

> **Note**
>
> IAM Identity Center does not work with SAMBA4-based Simple AD directories.

Attribute mappings between IAM Identity Center and External Identity Providers directory

Attribute mappings are used to map attribute types that exist in IAM Identity Center with like attributes in your external identity source such as Google Workspace, Microsoft Active Directory (AD), and Okta. IAM Identity Center retrieves user attributes from your identity source and maps them to IAM Identity Center user attributes.

If your IAM Identity Center is synchronized to use an **external identity provider** (IdP), like Google Workspace, Okta, or Ping as the identity source, you'll need to map your attributes in your IdP.

IAM Identity Center prefills a set of attributes for you under the **Attribute mappings** tab found on its 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 identity source. Each application determines the list of SAML 2.0 attributes it needs for successful single sign-on. 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 **Active Directory configuration page** if you're using Active Directory as an identity source. For more information, see Mapping user attributes between IAM Identity Center and Microsoft AD directory.

Supported external identity provider attributes

The following table lists all external identity provider (IdP) attributes supported and 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}
${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}

Supported attributes in your IdP
${path:enterprise.division}
${path:enterprise.department}
${path:enterprise.manager.value}

Default mappings between IAM Identity Center and Microsoft AD

The following table lists the default mappings for user attributes in IAM Identity Center to the user attributes in your Microsoft AD directory. IAM Identity Center only supports the list of attributes in the **User attribute in IAM Identity Center** column.

User attribute in IAM Identity Center	Maps to this attribute in your Active Directory
displayname	${displayname}
emails[?primary].value *	${mail}
externalid	${objectguid}
name.givenname	${givenname}
name.familyname	${sn}
name.middlename	${initials}
sid	${objectsid}
username	${userprincipalname}

* The email attribute in IAM Identity Center must be unique within the directory.

Group attribute in IAM Identity Center	Maps to this attribute in your Active Directory
`externalid`	`${objectguid}`
`description`	`${description}`
`displayname`	`${samaccountname}@{associat eddomain}`

Considerations

- If you do not have any assignments for your users and groups in IAM Identity Center when you enable configurable AD sync, the default mappings in the previous tables are used. For information about how to customize these mappings, see Configure attribute mappings for your sync.
- Certain IAM Identity Center attributes cannot 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.

Supported Microsoft AD attributes for IAM Identity Center

The following table lists all 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
`${samaccountname}`
`${description}`
`${objectguid}`

Supported attributes in your Microsoft AD directory
${objectsid}
${givenname}
${sn}
${initials}
${mail}
${userprincipalname}
${displayname}
${distinguishedname}
${proxyaddresses[?type == "SMTP"].value}
${proxyaddresses[?type == "smtp"].value}
${useraccountcontrol}
${associateddomain}

Considerations

- You can specify any combination of supported Microsoft AD directory attributes to map to a single mutable attribute in IAM Identity Center.

Supported IAM Identity Center attributes for Microsoft AD

The following table lists all IAM Identity Center attributes that are supported and that can be mapped to user attributes in your 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 for Active Directory
`${user:AD_GUID}`
`${user:AD_SID}`
`${user:email}`
`${user:familyName}`
`${user:givenName}`
`${user:middleName}`
`${user:name}`
`${user:preferredUsername}`
`${user:subject}`

Mapping user attributes between IAM Identity Center and 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 **`${mail}`**.
5. Choose **Save changes**.

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.

- 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.

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.

  - **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).

  - **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.

  - **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.

Topics

- How configurable AD sync works
- Configure attribute mappings for your sync
- First-time Active Directory to IAM Identity Center sync setup
- Add users and groups to your sync scope
- Remove users and groups from your sync scope
- Pause and resume your sync
- 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. To learn more about the prerequisites, see Prerequisites and considerations.

Creation

After you connect your self-managed directory in Active Directory or your AWS Managed Microsoft AD directory that is managed by 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.

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 attribute mappings for your sync

For more information about available attributes, see Attribute mappings between IAM Identity Center and External Identity Providers 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** cannot be changed.

6. Choose **Save changes**. This returns you to the **Manage Sync** page.

First-time Active Directory to IAM Identity Center sync setup

If you are synchronizing your users and groups from Active Directory into IAM Identity Center for the first time, follow these steps. Alternatively, you can follow steps outlined in Change your identity source to change your identity source from IAM Identity Center to Active Directory.

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

> **Note**
>
> When adding groups to your sync scope, sync groups directly from the trusted on-premises domain rather than from groups in the AWS Managed Microsoft AD domain. Groups synced directly from the trusted domain contain actual user objects that IAM Identity Center can access and synchronize successfully.

Add your Active Directory users and groups to IAM Identity Center by following these steps.

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**. If the user that you specified doesn't display 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**. If the group that you specified doesn't display 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.

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.

Manage users in the Identity Center directory

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 Directory Service. For more information, see considerations for Changing between IAM Identity Center directory 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
- Remove users from groups
- Edit Identity Center directory user properties

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. Alternatively, you can call the AWS API operation CreateUser to add users.

Console

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 cannot 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 emails with Amazon SES.
> 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 Provision users and groups from an external identity provider 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**.

AWS CLI

To add a user

The following `create-user` command creates a new user in your Identity Center directory.

```
aws identitystore create-user \
```

```
    --identity-store-id d-1234567890 \
    --user-name johndoe \
    --name "GivenName=John,FamilyName=Doe" \
    --display-name "John Doe" \
    --emails "Type=work,Value=johndoe@example.com"
```

Output:

```
{
    "UserId": "1234567890-abcdef",
    "IdentityStoreId": "d-1234567890"
}
```

> **Note**
>
> When you create users with the `create-user` CLI command or the CreateUser API operation, the users do not have passwords. You can update the settings in IAM Identity Center to send these users a verification email after their first attempt to sign on so they can set up a password. If you do not enable this setting, you must generate a one-time password and share it with the user. For more information, see Email one-time password to users created with API or CLI.

Add groups to your Identity Center directory

Use the following procedure to add groups to your Identity Center directory. Alternatively, you can call the AWS API operation CreateGroup to add groups.

Console

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**.

AWS CLI

To add a group

The following `create-group` command creates a new group in your Identity Center directory.

```
aws identitystore create-group \
    --identity-store-id d-1234567890 \
    --display-name "Developers" \
    --description "Group that contains all developers"
```

Output:

```
{
    "GroupId": "1a2b3c4d-5e6f-7g8h-9i0j-1k2l3m4n5o6p",
    "IdentityStoreId": "d-1234567890"
}
```

After you add this group to your Identity Center directory, you can assign single sign-on access to the group. For more information, see Assign user or group 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. Alternatively, you can call the AWS API operation CreateGroupMembership to add a user as a member of a group.

Console

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**.

AWS CLI

To add a user as a member of a group

The following `create-group-membership` command adds a user to a group in your Identity Center directory.

```
aws identitystore create-group-membership \
    --identity-store-id d-1234567890 \
    --group-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
    --member-id UserId=a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
    "MembershipId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "IdentityStoreId": "d-1234567890"
}
```

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.

> **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*.

Console

To delete a group

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**.

AWS CLI

To delete a group

The following `delete-group` command deletes the specified group from your Identity Center directory.

```
aws identitystore delete-group \
    --identity-store-id d-1234567890 \
    --group-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

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 you delete a user, you cannot undo this action. Use the following procedure to delete a user in your Identity Center directory.

> **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 Understanding authentication sessions 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*.

Console

To delete a user

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**.

AWS CLI

To delete a user

The following `delete-user` command deletes a user from your Identity Center directory.

```
aws identitystore delete-user \
    --identity-store-id d-1234567890 \
    --user-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Remove users from groups

Use the following procedure to remove members from a group. Alternatively, you can call the AWS API operation DeleteGroupMembership to remove a user from a group.

Console

To remove a user from a group

1. Open the IAM Identity Center console.
2. Choose **Groups**.
3. Choose the group you want to update.
4. On the group details page, under the **Users in this group**, choose the users to remove.
5. Choose **Remove users from group**.
6. On the **Remove users** dialog box, choose **Remove users from group** to verify you want to remove the users access to the account and applications that are assigned to the group.

AWS CLI

To remove a user from a group

The following `delete-group-membership` command removes a membership from a group.

```
aws identitystore delete-group-membership
    --identity-store-id d-1234567890 \
    --membership-id a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

Edit Identity Center directory user properties

Use the following procedure to edit the properties of a user in your Identity Center directory. Alternatively, you can call the AWS API operation UpdateUser to update user properties.

Console

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.

AWS CLI

To edit user properties in IAM Identity Center

The following `update-user` command updates the user's nickname.

```
aws identitystore update-user \
    --identity-store-id d-1234567890 \
    --user-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --operations '{"AttributePath":"nickName","AttributeValue":"Johnny"}'
```

Set up workforce access to AWS resources

Set up how your workforce users authenticate and access AWS resources through IAM Identity Center. This section covers the following components that govern workforce user access to your AWS environment:

- **Authentication sessions** – Understand how IAM Identity Center manages different types of user sessions, from interactive portal sessions to background application sessions, and how they interact with each other.
- **User access management** – Configure session durations, disable user accounts, and implement organization-wide access blocks to maintain security and compliance.
- **Password management** – For users created in the Identity Center directory, set password requirements, handle user credential setup, and manage password resets for users.
- **Multi-factor authentication** – For users created in the Identity Center directory, enhance security with MFA using authenticator apps, security keys, or built-in authenticators to protect user sign-ins.

Topics

- Understanding authentication sessions in IAM Identity Center
- Configure the session duration in IAM Identity Center
- Disable user access to AWS accounts and applications in IAM Identity Center
- Deny user access with Service Control Policies
- Managing access for users in the Identity Center directory

Understanding authentication sessions in IAM Identity Center

When a user signs in to the AWS access portal, IAM Identity Center creates an authentication session that represents the user's verified identity.

Once authenticated, the user can access all their assigned AWS accounts, AWS managed applications, and customer managed applications that administrators have granted them permission to use, without additional sign ins.

Types of authentication sessions

User interactive sessions

After a user signs in to the AWS access portal, IAM Identity Center creates a user interactive session. This session represents the user's authenticated state within IAM Identity Center and serves as the foundation for creating other session types. User interactive sessions can last for the duration configured in IAM Identity Center, which can be up to 90 days.

User interactive sessions are the primary authentication mechanism. They end when the user signs out or when an administrator ends their session. The duration of these sessions should be carefully configured based on your organization's security requirements.

For information about configuring user interactive session duration, see the section called "Configure session duration".

Application sessions

Application sessions are the authenticated connections between users and AWS managed applications (such as Kiro or Amazon Quick) that IAM Identity Center establishes through single sign-on.

By default, application sessions have a one hour lifetime, but they're automatically refreshed as long as the underlying user interactive session remains valid. This refresh mechanism provides a seamless experience for users while maintaining security controls. When a user interactive session ends, either through user sign-out or administrator action, the application sessions will end at their next refresh attempt, typically within 30 minutes.

User background sessions

User background sessions are extended-duration sessions designed for applications that need to run processes for hours or days without interruption. Currently, this session type applies primarily to Amazon SageMaker Studio, where data scientists might run machine learning training jobs that take many hours to complete.

For information about configuring user background session duration, see User background sessions.

Kiro sessions

You can extend Kiro sessions to allow developers using Kiro in IDEs to maintain authentication for up to 90 days. This feature reduces login interruptions while you work on code.

These sessions are independent of other session types and don't affect user interactive sessions or other AWS managed applications. Depending on when you enabled IAM Identity Center, this feature might be enabled by default.

For information about configuring extended Kiro sessions, see ???.

IAM Identity Center-created IAM role sessions

IAM Identity Center creates a different type of session when users access the AWS Management Console or AWS CLI. In these cases, IAM Identity Center uses the sign-in session to obtain an IAM session by assuming an IAM role specified in the user's permission set.

> **Important**
>
> Unlike application sessions, IAM role sessions operate independently once established. They persist for the duration configured in the permission set, which can be up to 12 hours, regardless of the status of the original sign-in session. This behavior ensures that long-running CLI operations or console sessions aren't unexpectedly ended.

Ways to end user sessions in IAM Identity Center

User-initiated

When a user signs out of the AWS access portal, the sign-in session ends, preventing the user from accessing any new resources.

Existing application sessions, however, don't end instantly. Instead, they will end within approximately 30 minutes, when they attempt their next refresh and find the sign-in session is no longer valid. Existing IAM role sessions continue until they expire based on the permission set configuration, which could be up to 12 hours later.

Administrator-initiated

Anyone with IAM Identity Center administrative permissions in your organization, typically IT administrators or security teams, can end a user's session. This action works the same way as

if users signed out themselves, allowing administrators to require users to sign in again when needed. This capability is useful when security policies change or when suspicious activity is detected.

When an IAM Identity Center administrator deletes a user or disables a user's access, the user loses access to the AWS access portal and is prevented from signing back in to start a new application or IAM role session. The user will lose access to existing application sessions within 30 minutes. Any existing IAM role sessions will continue based on the session duration configured in the IAM Identity Center permission set. The maximum session duration can be 12 hours.

What happens to user access when you end a session

This reference provides detailed information about how IAM Identity Center sessions behave when administrative actions are taken. The tables in this section show the duration and effects of user management actions and permission changes on access to the AWS access portal, applications, and AWS account sessions.

User management

This table summarizes how user management changes affect access to AWS resources, application sessions, and AWS account sessions.

Action	User loses IAM Identity Center access	User can't create new application sessions	User can't access existing application sessions	User loses access to existing AWS account sessions
User's access disabled	Effective immediately	Effective immediately	Within 30 minutes	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.
User deleted	Effective immediately	Effective immediately	Within 30 minutes	Within 12 hours or less. Duration

Action	User loses IAM Identity Center access	User can't create new application sessions	User can't access existing application sessions	User loses access to existing AWS account sessions
				depends on IAM role session expiry duration configured for the permission set.
User session revoked	User must sign in again to regain access	Effective immediately	Within 30 minutes	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.
User signs out	User must sign in again to regain access	Effective immediately	Within 30 minutes	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.

Group membership

This table summarizes how changes to user permissions and group memberships affect access to AWS resources, application sessions, and AWS account sessions.

Action	User loses IAM Identity Center access	User can't create new application sessions	User can't access existing application sessions	User loses access to existing AWS account sessions
Application or AWS account access removed from user	No - User can continue accessing IAM Identity Center	Effective immediately	Within 1 hour	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.
User removed from group that had an assigned application or AWS account	No - User can continue accessing IAM Identity Center	Within 1 hour	Within 2 hours	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.
Application or AWS account access removed from group	No - User can continue accessing IAM Identity Center	Effective immediately	Within 1 hour	Within 12 hours or less. Duration depends on IAM role session expiry duration configured for the permission set.

> **Note**
>
> The AWS access portal and AWS CLI will reflect updated user permissions within 1 hour after you add or remove a user from that group.

Understanding timing differences

- **Effective immediately** – Actions that require immediate re-authentication.
- **Within 30 minutes - 2 hours** – Application sessions need time to check with IAM Identity Center and discover any changes.
- **Within 12 hours or less** – IAM role sessions operate independently and only end when their configured duration expires.

Single logout

IAM Identity Center doesn't support SAML Single Logout (a protocol that automatically signs users out of all connected applications when they sign out of one) initiated by an identity provider that acts as your identity source. Additionally, it doesn't send SAML Single Logout to SAML 2.0 applications that use IAM Identity Center as an identity provider.

Best practices for session management

Effective session management requires thoughtful configuration and monitoring. Organizations should configure session durations appropriate to their security requirements, generally using shorter durations for sensitive applications and environments.

Implementing processes to end sessions when users change roles or leave the organization is essential for maintaining security boundaries. Regular review of active sessions should be incorporated into security monitoring practices to detect anomalous behavior that might indicate security issues, such as unusual access patterns, unexpected login times or locations, or access to resources outside normal job functions.

Configure the session duration in IAM Identity Center

You can configure the session duration for your workforce users when they use the AWS access portal and applications that work with IAM Identity Center, including Kiro. IAM Identity Center

provides the following session types: user interactive sessions, user background sessions, and extended sessions for Kiro.

Topics

- User interactive sessions
- User background sessions
- Extended sessions for Kiro
- View and end active sessions for your workforce users
- Session duration considerations for using identity sources, the AWS CLI, and AWS SDKs

User interactive sessions

User interactive sessions are sessions tied to a user's sign-in to the AWS access portal or access to AWS managed applications. The session duration of authentication into the AWS access portal and applications is the maximum length of time that a user can be signed in without re-authenticating. If you end an active AWS access portal session, this also ends any sessions for these managed applications.

The default session duration for user interactive sessions is 8 hours. You can specify a different duration, from a minimum of 15 minutes to a maximum of 90 days. Custom duration values must be entered in minutes and be between 15 minutes and 129,600 minutes (90 days). For more information, see Understanding authentication sessions in IAM Identity Center.

For considerations such as how IAM Identity Center identity sources might affect the user interactive session duration, see the section called “Considerations for identity sources, the AWS CLI, and AWS SDKs”.

To configure the duration of a user interactive session

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 duration**, choose **Configure**. A **Configure session duration** dialog box appears.

5. In the **Configure session duration** dialog box, under **User interactive sessions**, choose the maximum session duration for your users by selecting the drop-down arrow. Choose the length for the session, and then choose **Save**.

> **Note**
>
> Changes to session duration apply only to new sessions. Current sessions keep their original duration.

6. You are returned to the **Authentication** tab. A green notification message appears above the tab indicates that the session settings were updated successfully.

User background sessions

User background sessions allow a user to initiate a long-running job on an AWS managed application such as Amazon SageMaker Studio, without that user having to remain signed in while the job runs. The job runs immediately and uses the trusted identity propagation capability of IAM Identity Center to ensure that the user's permissions are maintained while the job is run in the background. The job can continue to run even if the user turns off their computer, their IAM Identity Center sign-in session expires, or the user signs out of the AWS access portal. This capability enables data scientists, machine learning engineers, and others to start analytics and machine learning workflows that run in the background without active user involvement.

User background sessions are enabled by default for supported AWS managed applications such as Amazon SageMaker Studio. To use this capability, however, you must enable trusted identity propagation in Amazon SageMaker Studio when you create or update a domain. For more information, see Enable trusted identity propagation in your Amazon SageMaker AI domain.

The default session duration for user background sessions is 7 days. You can specify a different duration, from a minimum of 15 minutes to a maximum of 90 days. Custom duration values must be entered in minutes and be between 15 minutes and 129,600 minutes (90 days).

Keep in mind the following considerations for user background sessions:

- A user background session can be created only when a user manually initiates a job in Amazon SageMaker Studio. This capability is not supported for automated, scheduled workflows.
- For a list of AWS Regions that support user background sessions, see Supported AWS Regions.

- You can view user background sessions in CloudTrail. For information, see Identifying user background session details.
- You can also end active sessions for a user in your organization. For information, see End active sessions for your workforce users.

To configure the duration of a user background session

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 duration**, choose **Configure**. The **Configure session duration** dialog box appears.
5. In the **Configure session duration** dialog box, if the **Enable user background sessions** check box is not already selected, select it. Clear the check box to disable user background sessions.

   > **Note**
   >
   > Current sessions are not affected if you disable user background sessions.

6. Under **User background sessions**, choose the maximum session duration by selecting the drop-down arrow. Choose the length for the session, and then choose **Save**.

   > **Note**
   >
   > Changes to session duration apply only to new sessions. Current sessions keep their original duration.

7. You are returned to the **Authentication** tab. A green notification message appears above the tab indicates that the session settings were updated successfully.

> **Note**
>
> A customer managed application can't create a user background session.

Extended sessions for Kiro

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

For considerations such as how IAM Identity Center identity sources might affect the extended session duration, see the section called "Considerations for identity sources, the AWS CLI, and AWS SDKs".

> **Note**
>
> Kiro 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 Kiro isn't currently accessible, enabling 90 day extended session duration won't override the default setting. This means that your session duration remains unchanged, whether you enable 90 day extended session duration or not. For information, Supported AWS Regions for Kiro.

To extend a session for Kiro

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 duration**, choose **Configure**. A **Configure session duration** dialog box appears.
5. In the **Configure session duration** dialog box, select the **Enable extended sessions for Kiro** check box. Clear the check box to disable extended session sessions for Kiro.
6. Choose **Save** to return to the **Settings** page.

View and end active sessions for your workforce users

As an IAM Identity Center administrator, you can view the list of your workforce users' active sessions, and if required, end one or more sessions for a user. For example, you might need to end a user's sessions when:

- The user no longer requires the sessions.
- The user shouldn't maintain their current authentication state. This can occur when they leave the company or their permissions change.

You can view and end these sessions by using the IAM Identity Center console. Your users can also view and end their own sessions by using the AWS access portal. For information about how your workforce users can view and end their sessions without assistance from an administrator, see Viewing and ending your active session.

> **Note**
>
> Ending an active session for an IAM Identity Center user doesn't end any active IAM role sessions in the AWS Management Console or AWS CLI. For more information, see Understanding authentication sessions in IAM Identity Center.

To end an active session for a workforce user (IAM Identity Center console)

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 active sessions for this user.
5. **Search for user background sessions (optional)**

   To search for sessions by the Amazon Resource Name (ARN) of the job that is using the session, in the **Session type** list, choose **User background sessions**, and then enter the job ARN in the search box.

   > **Note**
   >
   > You can only end active sessions that are loaded. If a user has many sessions, choose **Load more active sessions** to display additional sessions.

6. Select the check box next to each session that you want to end, and then choose **End sessions**.

7. A dialog box appears that confirms you are ending active sessions for this user. Review the information, and if you want to continue, type `confirm`, and then choose **End sessions**.
8. You are returned to the user's page. A green notification message appears to indicate that the selected sessions were successfully ended.

Session duration considerations for using identity sources, the AWS CLI, and AWS SDKs

Following are considerations for configuring the session duration if you use Microsoft Active Directory (AD) or an external identity provider (IdP) as the identity source, or the AWS Command Line Interface, AWS Software Development Kits (SDKs), or other AWS development tools to access AWS services programmatically.

Microsoft Active Directory, user interactive sessions, and extended sessions for Kiro

If you use Microsoft Active Directory (AD) as the identity source and you configure the session duration for user interactive sessions or extended sessions for Kiro, keep the following considerations in mind.

> **Note**
>
> These considerations do not apply to user background sessions.

Whether you use AWS Managed Microsoft AD or AD Connector configured in AWS Directory Service, the maximum lifetime for user Kerberos tickets defined in Microsoft AD can affect how long user interactive sessions and extended sessions for Kiro are valid. For more information about this setting, see Maximum lifetime for user ticket on the Microsoft website.

- **AWS Managed Microsoft AD**: If you use AWS Managed Microsoft AD configured in AWS Directory Service, the maximum lifetime for user Kerberos tickets is fixed at 10 hours. Therefore, the user interactive session duration is set to the shorter of the IAM Identity Center setting and 10 hours. For example, if you set the user interactive session duration to 12 hours, your users must re-authenticate in the AWS access portal after 10 hours. The same 10-hour limit applies to extended sessions for Kiro.

- **AD Connector**: If you use AD Connector configured in AWS Directory Service, the maximum lifetime for user Kerberos tickets is defined in Microsoft AD behind the AD Connector. The default value is 10 hours, and it has the same effect on user interactive sessions and extended sessions as for AWS Managed Microsoft AD. Although this limit might be configurable in Microsoft AD, we recommend that you work with your IT administrator to consider the risks, especially because this setting can affect the session duration for other Microsoft AD client applications.

External identity providers, user interactive sessions, and extended sessions for Kiro

If you use an external identity provider (IdP) and you configure the session duration for user interactive sessions or extended sessions for Kiro, keep the following considerations in mind.

> **Note**
>
> These considerations do not apply to user background sessions.

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

- If `SessionNotOnOrAfter` is not passed in a SAML assertion, the duration of an AWS access portal (user interactive) session and an extended 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. Similarly, if you set a 90-day extended session for Kiro, your Kiro users need to re-authenticate after 90 days.
- If `SessionNotOnOrAfter` is passed in a SAML assertion, the session duration value is set to the shorter of the AWS access portal (user interactive) session or extended 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. Similarly, if you set a 90-day extended session for Kiro, your Kiro users need to re-authenticate in Kiro after 18 hours.
- If the session duration of your IdP is longer than the one set in IAM Identity Center, your users can start a new IAM Identity Center session without re-entering their credentials, based on their still-valid login session with your IdP.

AWS CLI and SDK sessions

If you are using the AWS CLI, AWS 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 AWS managed 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

> **Note**
>
> For account access use cases, 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

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

When you disable user access in your IAM Identity Center directory, you cannot 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 disable user access for temporary access removal when you might need to restore access later.

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 will not be able

to create new sign in sessions. For more information, see Understanding authentication sessions in IAM Identity Center.

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 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 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**.

Deny user access with Service Control Policies

To immediately deny access to make authorized API calls when an IAM Identity Center user's access is disabled or the user is deleted, you can:

1. Add or update the inline policy of the permission set(s) assigned to the user by adding an explicit `Deny` effect for all actions on all resources.
2. Specify the `aws:userid` or `identitystore:userid` condition key.

Alternatively, you can use a Service Control Policy to deny the user's access across all member accounts in your organization.

Example SCP to deny access

This denial policy blocks all AWS actions for a specific user, regardless of other permissions they might have been granted elsewhere. This policy overrides any `Allow` policies.

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                 "StringLike": {
                    "aws:UserId": "*:deleteduser@domain.com"
                }
            }
        }
    ]
}
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                 "StringEquals": {
                    "identitystore:UserId": "DELETEDUSER_ID"
                }
            }
        }
    ]
}
```

Managing access for users in the Identity Center directory

Learn how to manage passwords and multi-factor authentication (MFA) for users in the IAM Identity Center directory. These security features help protect user accounts.

> **Note**
>
> These features do not apply to Active Directory users or external identity provider users.

Administrators can manage both passwords and MFA through the IAM Identity Center console. These security features work only with the built-in Identity Center directory.

Password management

Password management includes these capabilities:

- Reset passwords with email instructions
- Generate one-time passwords
- Configure automatic email verification for API-created users

AWS enforces fixed security requirements, including complexity rules and password reuse restrictions.

MFA

MFA is enabled by default and supports up to eight devices per user.

Supported device types include:

- Authenticator apps
- Security keys
- Built-in biometric authenticators

Administrators can register and manage MFA devices for users.

Topics

- Setting up user passwords

- MFA for Identity Center directory users

Setting up user passwords

For users created in the Identity Center directory, administrators can manage password policies, handle users without initial passwords, and reset passwords when needed. These password management features apply only to users in the built-in Identity Center directory. If you're using Active Directory or an external identity provider, you must manage passwords in those systems.

Password management options

- **Password requirements** – Security requirements that users must meet when setting or changing passwords. This includes complexity rules and reuse restrictions.
- **One-time password setup** – Configure email verification for users created through API or CLI who don't have initial passwords. You can also generate temporary passwords for immediate access.
- **Password resets** – Reset passwords for users who are locked out or need new credentials. You can send reset instructions using email or generate one-time passwords.

Topics

- Password requirements when managing identities in IAM Identity Center
- Email one-time password to users created with API or CLI
- Reset the IAM Identity Center user password for an end user

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.

Email one-time password to users created with API or CLI

When you create users with the CreateUser API operation or the `create-user` CLI command, the users do not have passwords. You can update the settings in IAM Identity Center to send these users a verification email after their first attempt to sign in, if you've specified an email for the user when they were created. After receiving the verification email, the user must set a password to sign in.

If you don't enable this setting, you must generate a one-time password and share it with users that you create using the CreateUser API or `create-user` CLI command.

To send an email address verification email to users created with the CreateUser API or `create-user` CLI command

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. In the **Configure standard authentication** dialog box, select the **Send email OTP** check box. Then, choose **Save**. The status updates from **Disabled** to **Enabled**.

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 the 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 are 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 cannot be reset from the IAM Identity Center console.
- **Users in the IAM Identity Center directory** – If you are 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.

MFA for Identity Center directory users

> **Important**
>
> MFA in IAM Identity Center is currently not supported for external identity providers.

IAM Identity Center comes preconfigured with multi-factor authentication (MFA) turned on by default so that all users must sign in with MFA in addition to their user name and password. This ensures that users must sign in to the AWS access portal using the following 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.

Topics

- Available MFA types for IAM Identity Center
- Configure MFA in IAM Identity Center
- Register an MFA device for users
- Renaming and deleting 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 AWS account. You can also configure MFA settings to require MFA whenever they attempt to sign-in from a new device or browser, or when signing in from an unknown IP address. For more information about how to configure MFA settings for your users, see Choose MFA types for user authentication 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 Feitian 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. 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 Directory Service, such as Amazon RDS for SQL Server.

If your MFA is **Disabled** on the IAM Identity Center console and you have configured RADIUS MFA with 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 in IAM Identity Center

You can configure Multi-factor authentication (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.

The following are general MFA recommendations, depending on your IAM Identity Center settings and organizational preferences.

- 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 will not 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 are already using RADIUS MFA that you configured with Directory Service, you do not 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.
- The following YouTube video provides an overview of MFA and IAM Identity Center:

IAM Identity Center: Multi-factor authentication defaults for new instances

Topics

- Prompt users for MFA
- Choose MFA types for user authentication
- Configure MFA device enforcement

- Allow users to register their own MFA devices

Prompt users for MFA

You can use the following steps to determine how often workforce users are prompted for multi-factor authentication (MFA) whenever they attempt to sign-in to the AWS access portal. Before you begin, we recommend that you understand the Available MFA types for IAM Identity Center.

> **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*.

> **Note**
>
> If you're using an external IdP, the **Multi-factor authentication** section will not be available. Your external IdP manages MFA settings, rather than IAM Identity Center managing them.

To configure 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:

   - **Every time they sign in (always-on)**

     In this mode (the default setting), IAM Identity Center requires that users with a registered MFA device will be prompted every time they sign in. This is the most secure setting and

ensures that your organizational or compliance policies are enforced by requiring that MFA be used 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.

- **Only when their sign-in context changes (context-aware)**

  In this mode, 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 but is less secure then the **always-on** option. Users are only prompted for MFA if their sign-in context changes.

- **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 and is not recommended.

  While MFA is disabled for your Identity Center directory for users, you cannot manage MFA devices in their user details, and Identity Center directory users cannot manage MFA devices from the AWS access portal.

  > **Note**
  >
  > If you are already using RADIUS MFA with 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 for user authentication

- Configure MFA device enforcement
- Allow users to register their own MFA devices

Choose MFA types for user authentication

Use the following procedure to choose the device types your users can authenticate with when prompted for multi-factor authentication (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.
   - **Security keys and built-in authenticators**
   - **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.

For more information about MFA in IAM, see AWS Multi-factor authentication in IAM.

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

IAM Identity Center administrators can allow 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.

Register an MFA device for users

IAM Identity Center administrators can set up a new MFA device for access by a specific user in the IAM Identity Center console. Administrators must have physical access to the user's MFA device 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

IAM Identity Center administrators can 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**.

Configure 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 Configure access to AWS accounts.

Topics

- AWS managed applications
- Customer managed applications
- Trusted identity propagation overview
- Setting up your own OAuth 2.0 application
- Rotate IAM Identity Center certificates
- Understand application properties in the IAM Identity Center console
- Assign user access to applications in the IAM Identity Center console
- Remove user access to SAML 2.0 applications
- Map attributes in your application to IAM Identity Center attributes

AWS managed applications

AWS IAM Identity Center streamlines and simplifies the task of connecting your workforce users to AWS managed applications such as Kiro and Amazon Quick. With IAM Identity Center, you can connect your existing identity provider once and synchronize users and groups from your directory, or create and manage your users directly in IAM Identity Center. By providing one point of federation, IAM Identity Center eliminates the need to set up federation or user and group synchronization for each application and reduces your administrative effort. You also get a common view of user and group assignments.

For a table of AWS applications that work with IAM Identity Center, see AWS managed applications that you can use with IAM Identity Center.

Controlling access to AWS managed applications

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. If you are 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.

  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.

  > **Important**
  >
  > 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.

- **Access to application resources**

  The application manages this through independent resource assignments that it controls.

AWS managed applications provide an administrative user interface that you can use to manage access to application resources. For example, Quick 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.

Sharing identity information

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.
- Alternatively, 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.

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.*

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 users in the Identity Center directory.
- 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 Microsoft AD directory.
  - **External identity provider** – For more information, see External identity providers.

  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.

Constraining the use of AWS managed applications

When you first enable IAM Identity Center, it becomes available as an identity source for AWS managed applications across all accounts in your AWS Organizations. To constrain applications, you must implement service control policies (SCPs). SCPs are a feature of AWS Organizations that you can use to centrally control the maximum permissions that identities (users and roles) in your organization can have. 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. For more information, see Service control policies (SCPs) in the *AWS Organizations User Guide.*

The following SCP example blocks access to the IAM Identity Center user and group information and prevents the application from being started, except in designated accounts (111111111111 and 222222222222):

```
{
  "Sid": "DenyIdCExceptInDesignatedAWSAccounts",
  "Effect": "Deny",
  "Action": [
    "identitystore:*",
    "sso:*",
    "sso-directory:*",
    "sso-oauth:*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "aws:PrincipalAccount": [
        "111111111111",
        "222222222222"
      ]
    }
  }
}
```

AWS managed applications that you can use with IAM Identity Center

IAM Identity Center lets you connect your existing identity source or create users once. This enables application administrators to manage access to the following AWS managed applications without separate federation or user and group synchronization.

All of the AWS managed applications in the following table integrate with organization instances of IAM Identity Center. The table also provides information about the following for a supported AWS managed application:

- Whether the application also integrates with account instances of IAM Identity Center
- Whether the application can enable trusted identity propagation through IAM Identity Center
- Whether the application supports IAM Identity Center configured with a customer managed KMS key
- Whether the application supports deployment in additional Regions of IAM Identity Center

> **Note**
>
> Applications that support deployment in additional Regions of IAM Identity Center also support IAM Identity Center configured with a customer managed KMS key. Every AWS managed application listed here supports deployment in the primary Region. For more information, see the section called "Deploying and managing AWS managed applications across multiple AWS Regions".

AWS managed applications that integrate with IAM Identity Center

AWS managed application	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center	Supports IAM Identity Center configured with a customer managed KMS key	Supports deployment in additional Regions of IAM Identity Center
Amazon Athena SQL	Yes	Yes	Yes	Yes
Amazon CodeCatalyst	Yes	No	Yes	No
Amazon DataZone	Yes	Yes	Yes	No
Amazon EKS Capabilities	Yes	Yes	Yes	No
Amazon EMR on EC2	Yes	Yes	Yes	No
Amazon EMR on EKS	Yes	Yes	Yes	No
Amazon EMR Serverless	Yes	Yes	Yes	No
Amazon EMR Studio	Yes	Yes	Yes	No
Amazon Kendra	No	No	Yes	No
Amazon Managed Grafana	No	No	Yes	No

AWS managed application	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center	Supports IAM Identity Center configured with a customer managed KMS key	Supports deployment in additional Regions of IAM Identity Center
Amazon Monitron	No	No	No	No
Amazon OpenSearch Service	Yes	Yes	Yes	No
Amazon OpenSearch Service Serverless Service	Yes	Yes	Yes	No
Amazon Q Business	Yes	Yes	Yes	No
Amazon Quick	Yes	Yes	Yes	No
Amazon Redshift	Yes[2]	Yes	Yes	Yes
Amazon S3 Access Grants	Yes	Yes	Yes	Yes
Amazon SageMaker Studio	No	Yes	Yes	No
Amazon SageMaker Unified Studio	Yes	Yes	Yes	Yes
Amazon WorkMail	Yes	Yes	Yes	No
Amazon WorkSpaces	Yes	No	Yes	No
Amazon WorkSpaces Secure Browser	No	No	Yes	No
AWS App Studio	Yes	No	No	No
AWS Deadline Cloud	Yes	No	Yes	Yes

AWS managed application	Integrated with account instances of IAM Identity Center	Enables trusted identity propagation through IAM Identity Center	Supports IAM Identity Center configured with a customer managed KMS key	Supports deployment in additional Regions of IAM Identity Center
AWS Glue	Yes	Yes	Yes	No
AWS IoT Events	No	No	No	No
AWS IoT SiteWise	No	No	No	No
AWS Lake Formation	Yes	Yes	Yes	Yes
AWS re:Post Private	Yes	No	Yes	No
AWS Supply Chain	Yes	No	Yes	No
AWS Systems Manager	No	No	Yes	Yes - Fleet Manager Remote Desktop
AWS Transfer Family web apps	Yes	Yes	Yes	Yes
AWS Transform	Yes	No	Yes	No
AWS Verified Access	No	No	Yes	No
Kiro	Yes[1]	Yes	Yes	No
Multi-party approval	No	Yes	Yes	No
OpenSearch user interface (Dashboards)	Yes	Yes	Yes	No

[1] For Kiro, account instances of IAM Identity Center are supported unless your users require access to the full set of Kiro features on AWS websites. For more information, see Setting up Kiro in the *Kiro User Guide*.

[2] For Amazon Redshift, account instances of IAM Identity Center are supported except for applications like Query Editor v2 that require permission sets, which are not supported by account instances.

> **Note**
>
> Some AWS services such as Amazon Connect and AWS Client VPN are not listed in this table although you can use them with IAM Identity Center. This is because they integrate with IAM Identity Center exclusively using SAML and are therefore categorized as customer managed applications.

Quick start: Setting up IAM Identity Center to test AWS managed applications

If your administrator hasn't already provided you with access to IAM Identity Center, you can use the steps in this topic to set up IAM Identity Center to test AWS managed applications. You'll learn how to enable IAM Identity Center, create a user directly in IAM Identity Center, and assign that user to an AWS managed application.

This topic provides quick-start steps on how to enable IAM Identity Center in either of the following ways:

- **With AWS Organizations** – If you choose this option, an *organization instance* of IAM Identity Center is created.
- **Only in your specific AWS account** – If you choose this option, an *account instance* of IAM Identity Center is created.

For information about these instance types, see Organization and account instances of IAM Identity Center.

Prerequisites

Before you enable IAM Identity Center, confirm the following:

- **You have an AWS account** – If you do not have an AWS account, see Getting started with an AWS account in the *AWS Account Management Reference Guide.*
- **The AWS managed application works with IAM Identity Center** – Review the list of AWS managed applications that you can use with IAM Identity Center to confirm that the AWS managed application you want to test works with IAM Identity Center.
- **You've reviewed Regional considerations** – Make sure that the AWS managed application you want to test is supported in the AWS Region where you enable IAM Identity Center. For more information, see the documentation for the AWS managed application.

> **Note**
>
> You must deploy your AWS managed application in the same Region where you plan to enable IAM Identity Center.

Setting up an organization instance of IAM Identity Center to test AWS managed applications

> **Note**
>
> This topic describes how to enable IAM Identity Center with AWS Organizations, which is the recommended way to enable IAM Identity Center.

Confirm your permissions

To enable IAM Identity Center with AWS Organizations, you must sign in to the AWS Management Console as either of the following:

- A user with administrative permissions in the AWS account where IAM Identity Center will be enabled with AWS Organizations.
- The root user (not recommended unless no other administrative users exist).

> **Important**
>
> The root user has access to all AWS services and resources in the account. As a security best practice, unless you have no other credentials, do not use your account's root

credentials to access AWS resources. These credentials provide unrestricted account access and are difficult to revoke.

Step 1. Enable IAM Identity Center with AWS Organizations

1. Do one 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 with a standalone AWS account (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. On the AWS Management Console Home page, select the IAM Identity Center service or navigate to the IAM Identity Center console.
3. Choose **Enable**, and enable IAM Identity Center with AWS Organizations. When you do this, you're creating an organization instance of IAM Identity Center.

Step 2. Create an administrative user in IAM Identity Center

This procedure describes how to create a user directly in the built-in Identity Center directory. This directory isn't connected to any other directory that your administrator might use to manage workforce users. After you create the user in IAM Identity Center, you'll specify new credentials for this user. When you sign in as this user to test your AWS managed application, you'll sign in with the new credentials, not with any existing credentials that you use to access corporate resources.

> **Note**
>
> We recommend that you use this method for creating users for testing purposes only.

1. In the navigation pane of the IAM Identity Center console, choose **Users**, and then choose **Add user**.
2. Follow the guidance in the console to add the user. Keep **Send an email to this user with password setup instructions** selected and make sure that you specify an email address to which you have access.
3. In the navigation pane, choose AWS accounts, select the check box next to your account, and choose **Assign users or groups**.

4. Choose the **Users** tab, select the check box next to the user that you just added, and choose **Next**.
5. Choose **Create permission set**, and follow the guidance in the console to create the `AdministratorAccess` predefined permission set.
6. When you're done, the new permission set appears in the list. Close the **Permission sets** tab in your browser window, return to the **Assign users and groups** tab, and choose the refresh icon next to **Create permission set**.
7. On the **Assign users and groups** browser tab, the new permission set appears in the list. Select the check box next to the name of the permission set, choose **Next**, and then choose **Submit**.
8. Sign out of the console.

Step 3. Sign in to the AWS access portal as an administrative user

The AWS access portal is a web portal that provides the user that you created with access to the AWS Management console. Before you can sign in to the AWS access portal, you must accept the invitation to join IAM Identity Center and activate your user credentials.

1. Check your email for the subject line **Invitation to join AWS IAM Identity Center**.
2. Choose **Accept invitation**, and follow the guidance on the sign-up page to set a new password, sign in, and register an MFA device for your user.
3. After you register your MFA device, the AWS access portal opens.
4. In the AWS access portal, select your AWS account and choose **AdministratorAccess**. You are redirected to the AWS Management Console.

Step 4. Configure the AWS managed application to use IAM Identity Center

1. While you are signed in to the AWS Management Console, open the console for the AWS managed application that you plan to use.
2. Follow the guidance in the console to configure the AWS managed application to use IAM Identity Center. During this process, you can assign the user that you created to the application.

Setting up an account instance of IAM Identity Center to test AWS managed applications

> **Note**
>
> An account instance of IAM Identity Center limits your deployment to a single AWS account. You must enable this instance in the same AWS Region as the AWS application you want to test.

Confirm your app

All AWS managed applications that work with IAM Identity Center can be used with organization instances of IAM Identity Center. However, only some of these applications can be used with account instances of IAM Identity Center. Review the list of AWS managed applications that you can use with IAM Identity Center.

Step1. Enable an account instance of IAM Identity Center

1. Do one 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 with a standalone AWS account (IAM credentials)** – Sign in using your IAM credentials with administrative permissions.
2. On the AWS Management Console Home page, select the IAM Identity Center service or navigate to the IAM Identity Center console.
3. Choose **Enable**.
4. On the **Enable IAM Identity Center with AWS Organizations** page, choose **enable an account instance of IAM Identity Center**.
5. On the **Enable account instance of IAM Identity Center** page, review the information and optionally add tags that you want to associate with this account instance. Then choose **Enable**.

Step 2. Create a user in IAM Identity Center

This procedure describes how to create a user directly in the built-in Identity Center directory. This directory isn't connected to any other directory that your administrator might use to manage

workforce users. After you create the user in IAM Identity Center, you'll specify new credentials for this user. When you sign in as this user to test your AWS managed application, you'll sign in with the new credentials. The new credentials will not allow you to access other corporate resources.

> **Note**
>
> We recommend that you use this method for creating users for testing purposes only.

1. In the navigation pane of the IAM Identity Center console, choose **Users**, and then choose **Add user**.
2. Follow the guidance in the console to add the user. Keep **Send an email to this user with password setup instructions** selected and make sure that you specify an email address to which you have access.
3. Sign out of the console.

Step 3. Sign in to the AWS access portal as your IAM Identity Center user

The AWS access portal is a web portal that provides the user that you created with access to the AWS Management console. Before you can sign in to the AWS access portal, you must accept the invitation to join IAM Identity Center and activate your user credentials.

1. Check your email for the subject line **Invitation to join AWS IAM Identity Center**.
2. Choose **Accept invitation**, and follow the guidance on the sign-up page to set a new password, sign in, and register an MFA device for your user.
3. After you register your MFA device, the AWS access portal opens. When applications are available to you, you'll find them under the **Applications** tab.

   > **Note**
   >
   > AWS applications that support account instances allow users to sign in to applications without requiring additional permissions. Therefore, the **Accounts** tab will remain empty.

Step 4. Configure the AWS managed application to use IAM Identity Center

1. While you are signed in to the AWS Management Console, open the console for the AWS managed application that you plan to use.
2. Follow the guidance in the console to configure the AWS managed application to use IAM Identity Center. During this process, you can assign the user that you created to the application.

Viewing and changing 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 change details about the application in the IAM Identity Center console.

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 overview.

To view and change 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. If you want to change information about an AWS managed application, 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.

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**.

> **Note**
>
> If an AWS managed application is disabled, you can restore users abilty to authenticate to the application by choosing **Actions** and then **Enable**.

Enabling identity-enhanced console sessions

An identity-enhanced 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 Kiro Pro users of Kiro on AWS apps and websites.

You can enable identity-enhanced 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-enhanced-console sessions
- How identity-enhanced console sessions work

Prerequisites and considerations

Before you enable identity-enhanced console sessions, review the following prerequisites and considerations:

- If your users access Kiro on AWS apps and websites through an Kiro Pro subscription, you must enable identity-enhanced console sessions.

> **Note**
>
> Kiro users can access Kiro without identity-enhanced sessions, but they will not have access to their Kiro Pro subscriptions.

- Identity-enhanced console sessions require an organization instance of IAM Identity Center.
- Integration with Kiro isn't supported if you enable IAM Identity Center in an opt-in AWS Region.
- To enable identity-enhanced 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-enhanced console sessions, you must grant them the `sts:setContext` permission in an identity-based policy. For information, see Granting users permissions to use identity-enhanced console sessions.

How to enable identity-enhanced-console sessions

You can enable identity-enhanced console sessions in the Kiro console or in the IAM Identity Center console.

Enable identity-enhanced console sessions in the Kiro console

Before you enable identity-enhanced 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 Enable 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 are not already using another identity source. For more information, see IAM Identity Center identity source tutorials.
3. After you finish setting up IAM Identity Center, open the Kiro console and follow the steps in Subscriptions in the *Kiro User Guide*. Make sure to enable identity-enhanced console sessions.

> **Note**
>
> If you do not have sufficient permissions to enable identity-enhanced 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-enhanced console sessions in the IAM Identity Center console

If you are an IAM Identity Center administrator, you might be asked by another administrator to enable identity-enhanced 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-enhanced sessions**, choose **Enable**.
4. In the second message, choose **Enable**.
5. After you finish enabling identity-enhanced console sessions, a confirmation message appears at the top of the **Settings** page.

6. In the **Details** section, the status for **Identity-enhanced sessions** is **Enabled**.

How identity-enhanced 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-enhanced 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.

Customer managed applications

IAM Identity Center acts as a central identity service to your workforce users and groups. 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. With a single connection, IAM Identity Center represents your IdP in front of multiple AWS services and enables your OAuth 2.0 applications to request access to data in these services on behalf of your users. You can also use IAM Identity Center to assign your users access to SAML 2.0

applications. This includes AWS services such as Amazon Connect and AWS Client VPN, which integrate with IAM Identity Center exclusively using SAML and are therefore categorized as customer managed applications.

- If your application supports **JSON Web Tokens (JWTs)**, you can use the trusted identity propagation feature of IAM Identity Center to enable your application to request access to data in AWS services on behalf of your users. Trusted identity propagation is built on the OAuth 2.0 Authorization Framework and includes an option for applications to exchange identity tokens that come from an external OAuth 2.0 authorization server for tokens issued by IAM Identity Center and recognized by AWS services. For more information, see Trusted identity propagation use cases.
- If your application supports **SAML 2.0**, you can connect it to an organization instance of IAM Identity Center. You can use IAM Identity Center to assign access to your SAML 2.0 application.

> **Note**
>
> When integrating customer managed applications with an IAM Identity Center instance that uses a customer managed KMS key, verify whether the application invokes IAM Identity Center service APIs. If it does, see the baseline KMS key policy for the required permissions.

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.

> **Note**
>
> IAM Identity Center does not support validating signatures of incoming SAML authentication requests from SAML applications.

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.

> **Note**
>
> If you have clients using `device.sso.region.amazonaws.com`, you must update your authorization flow to use Proof Key for Code Exchange (PKCE). For more information, see Configuring IAM Identity Center authentication with the AWS CLI in the *AWS Command Line Interface User Guide*.

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 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
`sso:account:access`	Access IAM Identity Center managed accounts and permission sets.	IAM Identity Center
`codewhisperer:analysis`	Enable access to Kiro code analysis.	AWS Builder ID and IAM Identity Center
`codewhisperer:completions`	Enable access to Kiro inline code suggestions.	AWS Builder ID and IAM Identity Center
`codewhisperer:conversations`	Enable access to Kiro chat.	AWS Builder ID and IAM Identity Center
`codewhisperer:taskassist`	Enable access to Kiro Agent for software development.	AWS Builder ID and IAM Identity Center

Scope	Description	Services supported by
`codewhisperer:transformations`	Enable access to Kiro Agent for code transformation.	AWS Builder ID and IAM Identity Center
`codecatalyst:read_write`	Read and write to your Amazon CodeCatalyst resources, allowing access to all your existing resources.	AWS Builder ID and IAM Identity Center
`verified_access:application:connect`	Enable AWS Verified Access	AWS Verified Access
`redshift:connect`	Connect to Amazon Redshift	Amazon Redshift
`datazone:domain:access`	Access your DataZone Domain Execution Role	Amazon DataZone
`nosqlworkbench:datamodeladviser`	Create and read data models	NoSQL Workbench
`transform:read_write`	Enable access to AWS Transform Agent for code transformation	AWS Transform

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.

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 is helpful to have the service provider's metadata exchange file so that you can more efficiently set up the trust. If you do not 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 Understand 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 do not 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 Understand 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 overview

Trusted identity propagation is a feature of IAM Identity Center that enables administrators of AWS services to grant permissions based on user attributes such as group associations. With trusted identity propagation, identity context is added to an IAM role to identify the user requesting access to AWS resources. This context is propagated to other AWS services.

Identity context comprises information that AWS services use to make authorization decisions when they receive access requests. This information includes metadata that identifies the requester (for example, an IAM Identity Center user), the AWS service to which access is requested (for example, Amazon Redshift), and the scope of access (for example, read only access). The receiving AWS service uses this context, and any permissions assigned to the user, to authorize access to its resources.

Benefits of trusted identity propagation

Trusted identity propagation allows the administrators of AWS services to grant permissions to resources, such as data, using the corporate identities of your workforce. In addition, they can audit who accessed what data by looking at service logs or AWS CloudTrail. If you are an IAM Identity Center administrator, you may be asked by other AWS service administrators to enable trusted identity propagation.

Enabling trusted identity propagation

The process of enabling trusted identity propagation involves the following two steps:

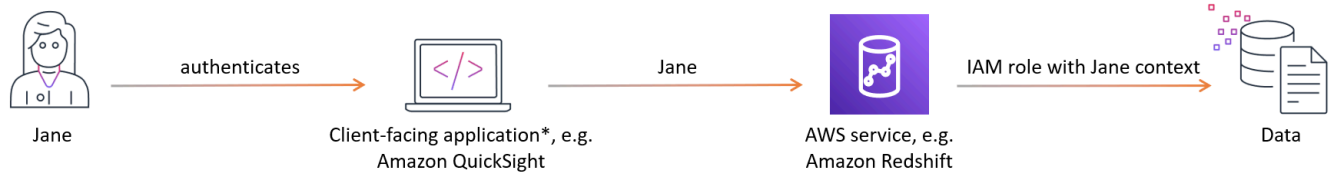
1. **Enable IAM Identity Center and connect your existing source of identities to IAM Identity Center** - You'll continue to manage your workforce identities in your existing source of identities; connecting it to IAM Identity Center creates a reference to your workforce that all AWS services in your use case can share. It's also available for data owners to use in future use cases.

2. **Connect the AWS services in your use case to IAM Identity Center** - The administrator of each AWS service in the trusted identity propagation use case follows the guidance in the respective service documentation to connect the service to IAM Identity Center.

> **Note**
>
> If your use case involves a *third-party* or *customer developed application*, you enable trusted identity propagation by configuring a trust relationship between the identity provider that authenticates the application users and IAM Identity Center. This allows your application to take advantage of the trusted identity propagation flow previously described.
> For more information, see Using applications with a trusted token issuer.

How trusted identity propagation works

The following diagram shows the high-level workflow for trusted identity propagation:

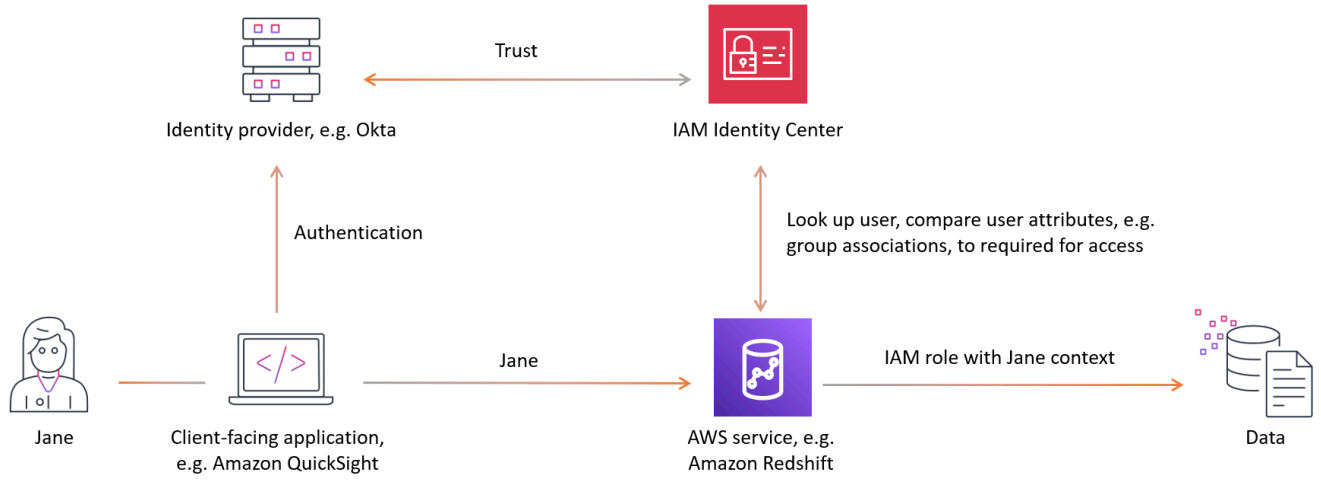


1. Users authenticate with a client-facing application, for example Quick.
2. The client-facing application requests access to use an AWS service to query data and includes information on the user.

> **Note**
>
> Some trusted identity propagation use cases involve tools that interact with AWS services using service drivers. You can find out if this applies to your use case in the use case guidance.

3. The AWS service verifies the user identity with IAM Identity Center and compares the user attributes, like their group associations, with those required for access. The AWS service authorizes the access so long as the user or their group has the necessary permissions.
4. AWS services may log the user identifier in AWS CloudTrail and in their service logs. Check the service documentation for details.

The following image provides an overview of the previously described steps in the trusted identity propagation workflow:



Topics

- Prerequisites and considerations
- Trusted identity propagation use cases
- Authorization services

Prerequisites and considerations

Before you set up trusted identity propagation, review the following prerequisites and considerations.

Topics

- Prerequisites
- Considerations
- Considerations for customer managed applications

Prerequisites

To use trusted identity propagation, ensure your environment meets the following prerequisites:

- Enable and provision IAM Identity Center

- To use trusted identity propagation, you must enable IAM Identity Center in the same AWS Region where the AWS applications and services your users will access are enabled. For information, see Enable IAM Identity Center.
  - IAM Identity Center Organization instance is recommended - We recommend you use an organization instance of IAM Identity Center that you enable in the management account of AWS Organizations. You can delegate administration of an organization instance of IAM Identity Center to a member account. If you choose an account instance of IAM Identity Center, all AWS services that you want users to access with trusted identity propagation must reside in the same AWS account where you enable IAM Identity Center. For more information, see Account instances of IAM Identity Center.
- Connect your existing identity provider to IAM Identity Center and provision your users and groups into IAM Identity Center. For more information, see IAM Identity Center identity source tutorials.
- Connect the AWS managed applications and services in your trusted identity propagation use case to IAM Identity Center. To use trusted identity propagation, AWS managed applications must be connected to IAM Identity Center.

Considerations

Keep in mind the following considerations when configuring and using trusted identity propagation:

- **Organization vs account instance of IAM Identity Center**
  - An organization instance of IAM Identity Center will give you the most control and flexibility to grow your use cases to multiple AWS accounts, users, and AWS services. If you are unable to use an organization instance, your use case may be supported with account instances of IAM Identity Center. To learn more about which AWS services in your use case support account instances of IAM Identity Center, see AWS managed applications that you can use with IAM Identity Center.
- **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.

Considerations for customer managed applications

Your workforce can benefit from trusted identity propagation even if your users interact with client-facing applications that are not managed by AWS, for example Tableau or your custom-developed applications. The users of these applications may not be provisioned in IAM Identity Center. To enable the smooth recognition and authorization of user access to AWS resources, IAM Identity Center enables you to configure a trusted relationship between the identity provider authenticating your users and IAM Identity Center. For more information, see Using applications with a trusted token issuer.

In addition, configuring trusted identity propagation for your application will require:

- Your application must use OAuth 2.0 framework for authentication. Trusted identity propagation does not support SAML 2.0 integrations.
- Your application must be recognized by IAM Identity Center. Follow the guidance specific to your use case.

Trusted identity propagation use cases

As an IAM Identity Center administrator, you might be asked to help configure trusted identity propagation from user facing applications to AWS services. To support this request, you'll need the following information:

- What client-facing application will your users interface with?
- Which AWS services are used to query the data and to authorize access to the data?
- Which AWS service authorizes access to the data?

Your role in enabling **trusted identity propagation use cases that do not involve third-party applications or custom-developed applications** is to:

1. Enable IAM Identity Center.
2. Connect your existing source of identities to IAM Identity Center.

The remaining steps of the trusted identity configuration for these use cases are performed within the connected AWS services and applications. The administrators of the connected AWS services or applications should refer to the respective user guides for comprehensive service-specific guidance.

Your role in enabling **trusted identity propagation use cases that involve third-party applications or custom-developed applications** includes the steps of Enable IAM Identity Center and connecting your source of identities as well as:

1. Configuring the connection of your identity provider (IdP) to the third-party party or custom-developed application.
2. Enabling IAM Identity Center to recognize the third-party or custom-developed application.
3. Configuring your IdP as a trusted token issuer in IAM Identity Center. For more information, see Using applications with a trusted token issuer.

The administrators of the connected applications and AWS services should refer to the respective user guides for comprehensive service-specific guidance.

Analytics, data lakehouse, and machine learning use cases

You can enable trusted propagation use cases with the following analytics and machine learning services:

- **Amazon Redshift** - For guidance, see Trusted identity propagation with Amazon Redshift.
- **Amazon EMR** - For guidance, see Trusted identity propagation with Amazon EMR.
- **Amazon Athena** - For guidance, see Trusted identity propagation with Amazon Athena.
- **SageMaker Studio** - For guidance, see Trusted identity propagation with Amazon SageMaker Studio.

Additional use cases

You can enable IAM Identity Center and trusted identity propagation with these additional AWS services:

- **Amazon Q Business** - for guidance, see:
  - Admin workflow for apps using IAM Identity Center.
  - Configuring an Amazon Q Business application using IAM Identity Center.
  - Configure Amazon Q Business with IAM Identity Center trusted identity propagation.
- **Amazon OpenSearch Service** - for guidance, see:
  - IAM Identity Center Trusted Identity Propagation Support for Amazon OpenSearch Service.

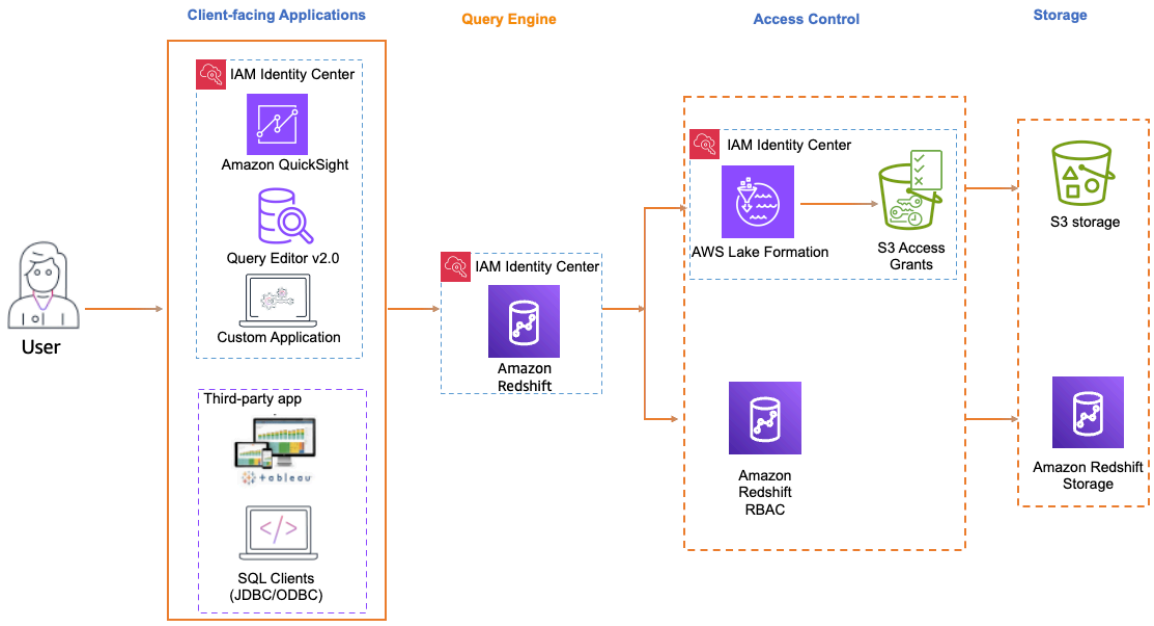
- Centralized OpenSearch user interface (Dashboards) with Amazon OpenSearch Service.
- **AWS Transfer Family** - for guidance, see:
  - Transfer Family web apps.

Topics

- Trusted identity propagation with Amazon Redshift
- Trusted identity propagation with Amazon EMR
- Trusted identity propagation with Amazon Athena
- Trusted identity propagation with Amazon SageMaker Studio

Trusted identity propagation with Amazon Redshift

The steps to enable trusted identity propagation depend on whether your users interact with AWS managed applications or customer managed applications. The following diagram shows a trusted identity propagation configuration for client-facing applications - either AWS managed or external to AWS - that query Amazon Redshift data with access control provided either by Amazon Redshift or by authorization services, such as AWS Lake Formation or Amazon S3 Access Grants.



When trusted identity propagation to Amazon Redshift is enabled, Redshift administrators can configure Redshift to automatically create roles for IAM Identity Center as the identity provider, map Redshift roles to groups in IAM Identity Center, and use Redshift role-based access control to grant access.

Supported client-facing applications

AWS managed applications

The following AWS managed client-facing applications support trusted identity propagation to Amazon Redshift:

- Amazon Redshift Query Editor V2
- Quick

> **Note**
>
> If you are using Amazon Redshift Spectrum to access external databases or tables in AWS Glue Data Catalog, consider setting up Lake Formation and Amazon S3 Access Grants to provide fine-grain access control.

Customer managed applications

The following customer managed applications support trusted identity propagation to Amazon Redshift:

- **Tableau** including Tableau Desktop, Tableau Server, and Tableau Prep
  - To enable trusted identity propagation for users of Tableau, refer to Integrate Tableau and Okta with Amazon Redshift using IAM Identity Center in the *AWS Big Data Blog*.
- **SQL Clients** (DBeaver and DBVisualizer)
  - To enable trusted identity propagation for users of SQL Clients (DBeaver and DBVisualizer), refer to Integrate Identity Provider (IdP) with Amazon Redshift Query Editor V2 and SQL Client using IAM Identity Center for seamless Single Sign-On in the *AWS Big Data Blog*.

Setting up trusted identity propagation with Amazon Redshift Query Editor V2

The following procedure walks you through how to achieve trusted identity propagation from Amazon Redshift Query Editor V2 to Amazon Redshift.

Prerequisites

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

1. Enable IAM Identity Center. Organization instance is recommended. For more information, see Prerequisites and considerations.
2. Provision the users and groups from your source of identities into IAM Identity Center.

Enabling trusted identity propagation includes tasks performed by an IAM Identity Center administrator in the IAM Identity Center console and tasks performed by an Amazon Redshift administrator in the Amazon Redshift console.

Tasks performed by the IAM Identity Center administrator

The following tasks needed to be complete by the IAM Identity Center administrator:

1. **Create an IAM role** in the account where the Amazon Redshift cluster or Serverless instance exists with the following permission policy. For more information, see IAM Role creation.
   - The following policy examples includes the necessary permissions to complete this tutorial. To use this policy, replace the *italicized placeholder text* in the example policy with your own information. For additional directions, see Create a policy or Edit a policy.

     Permission policy:

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRedshiftApplication",
            "Effect": "Allow",
            "Action": [
                "redshift:DescribeQev2IdcApplications",
                "redshift-serverless:ListNamespaces",
                "redshift-serverless:ListWorkgroups",
                "redshift-serverless:GetWorkgroup"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIDCPermissions",
```

```
            "Effect": "Allow",
            "Action": [
                "sso:DescribeApplication",
                "sso:DescribeInstance"
            ],
            "Resource": [
                "arn:aws:sso:::instance/Your-IAM-Identity-Center-Instance
 ID",
                "arn:aws:sso::111122223333:application/Your-IAM-Identity-
Center-Instance-ID/*"
            ]
        }
    ]
}
```

Trust policy:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "redshift-serverless.amazonaws.com",
                    "redshift.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ]
        }
    ]
}
```

2. **Create a permission set** in the AWS Organizations management account where IAM Identity Center is enabled. You'll use it in the next step to allow federated users to access Redshift Query Editor V2.

   a. Go to the **IAM Identity Center** console, under **Multi-Account permissions**, choose **Permission sets**.

   b. Choose **Create permission set**.

   c. Choose **Custom permission set** and then choose **Next**.

   d. Under **AWS managed policies**, choose **`AmazonRedshiftQueryEditorV2ReadSharing`**.

   e. Under **Inline policy**, add the following policy:

      JSON

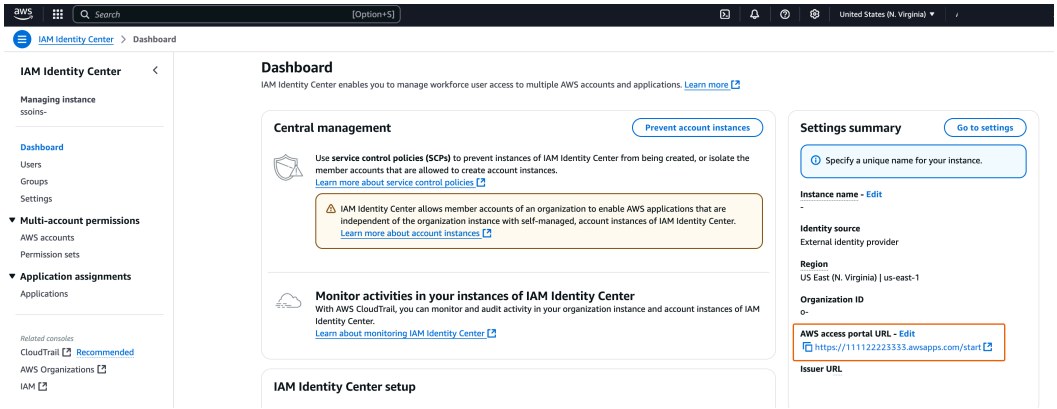
```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "redshift:DescribeQev2IdcApplications",
                "redshift-serverless:ListNamespaces",
                "redshift-serverless:ListWorkgroups",
                "redshift-serverless:GetWorkgroup"
            ],
            "Resource": "*"
        }
    ]
}
```

   f. Select **Next** and then provide a name for the permission set name. For example, **`Redshift-Query-Editor-V2`**.

   g. Under **Relay state – optional**, set default relay state to the Query Editor V2 URL, using the format: `https://`*`your-region`*`.console.aws.amazon.com/sqlworkbench/home`.

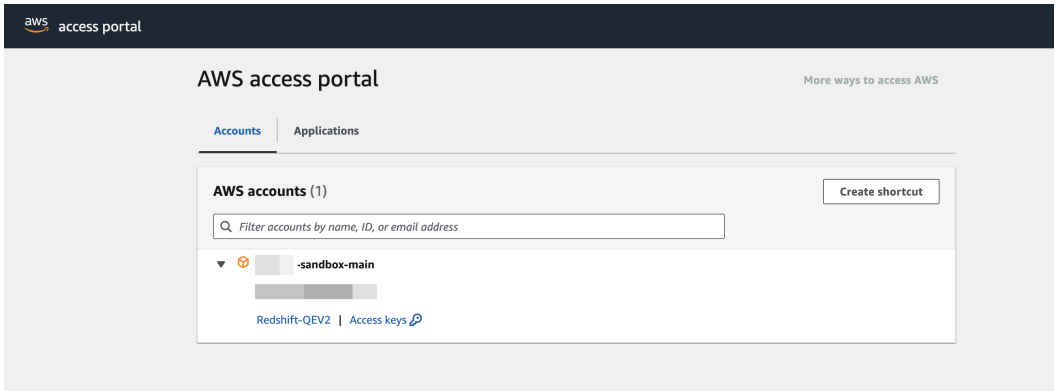
   h. Review the settings and choose **Create**.

   i. Navigate to the IAM Identity Center Dashboard and copy the AWS access portal URL from the **Setting Summary** section.



j. Open a new Incognito Browser Window and paste the URL.

This will take you to your AWS access portal, ensuring you are signing in with an IAM Identity Center user.



For more information about permission set, see Manage AWS accounts with permission sets.

3. **Enable federated users access to Redshift Query Editor V2**.

   a. In the AWS Organizations management account, open the **IAM Identity Center** console.

   b. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.

   c. On the AWS accounts page, select the AWS account that you want to assign access to.

   d. Choose **Assign users or groups**.

   e. On the **Assign users and groups** page, choose the users and or groups that you want to create the permission set for. Then, choose **Next**.

   f. On the **Assign permission sets** page, choose the permission set you created in the previous step. Then, choose **Next**.

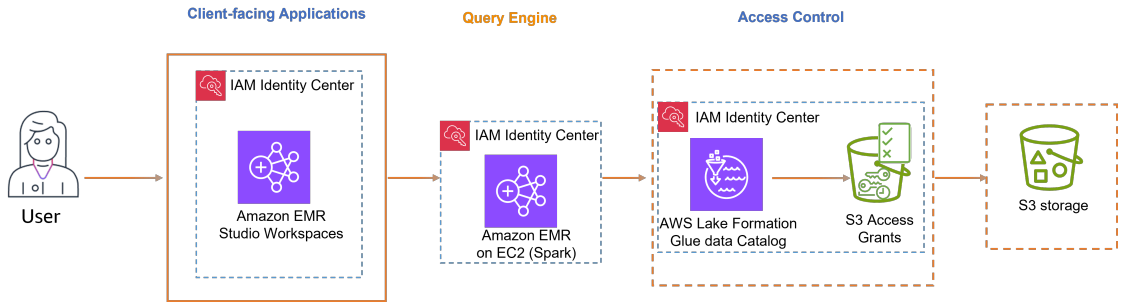
   g. On the **Review and submit assignments** page, review your selections and choose **Submit**.

Tasks performed by an Amazon Redshift administrator

Enabling trusted identity propagation to Amazon Redshift requires an Amazon Redshift cluster administrator or Amazon Redshift Serverless administrator to perform a number of tasks in the Amazon Redshift console. For more information, see Integrate Identity Provider (IdP) with Amazon Redshift Query Editor V2 and SQL Client using IAM Identity Center for seamless Single Sign-On in the *AWS Big Data Blog*.

Trusted identity propagation with Amazon EMR

The following diagram shows a trusted identity propagation configuration for Amazon EMR Studio using Amazon EMR on Amazon EC2 with access control provided by AWS Lake Formation and Amazon S3 Access Grants.



Supported client-facing applications

- Amazon EMR Studio

To enable trusted identity propagation, follow these steps:

- Set up Amazon EMR Studio as the client-facing application for Amazon EMR cluster.
- Set up Amazon EMR Cluster on Amazon EC2 with Apache Spark.
- *Recommended*: AWS Lake Formation and Amazon S3 Access Grants to provide fine-grained access control to AWS Glue Data Catalog and underlying data locations in S3.

Setting up trusted identity propagation with Amazon EMR Studio

The following procedure walks you through setting up Amazon EMR Studio for trusted identity propagation in queries against an Amazon Athena workgroups or Amazon EMR clusters running Apache Spark.

Prerequisites

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

1. Enable IAM Identity Center. Organization instance is recommended. For more information, see Prerequisites and considerations.
2. Provision the users and groups from your source of identities into IAM Identity Center.

To complete setting up trusted identity propagation from Amazon EMR Studio, the EMR Studio administrator must perform the following steps.

Step 1. Create the required IAM roles for EMR Studio

In this step, the Amazon EMR Studio administrator creates and IAM service role and an IAM user role for EMR Studio.

1. **Create an EMR Studio service role** - EMR Studio assume this IAM role to securely manage workspaces and notebooks, connect to clusters, and handle data interactions.

   a. Navigate to the IAM console (https://console.aws.amazon.com/iam/) and create an IAM role.

   b. Select **AWS service** as the trusted entity and then choose **Amazon EMR**. Attach the following policies to define the role's permissions and trust relationship.

      To use these policy, replace the *italicized placeholder text* in the example policy with your own information. For additional directions, see Create a policy or Edit a policy.

      JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ObjectActions",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::Your-S3-Bucket-For-EMR-Studio/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "Your-AWS-Account-ID"
                }
            }
        },
        {
            "Sid": "BucketActions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::Your-S3-Bucket-For-EMR-Studio"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "Your-AWS-Account-ID"
                }
            }
        }
    ]
}
```

For a reference of all the service role permissions, see EMR Studio service role permissions.

2. **Create an EMR Studio user role for IAM Identity Center authentication** - EMR Studio assumes this role when a user signs in through IAM Identity Center to manage workspaces, EMR clusters, jobs, git repositories. **This role is used to initiate the trusted identity propagation workflow**.

> **Note**
>
> The EMR Studio user role does not need to include permissions to access the Amazon S3 locations of the tables in AWS Glue Catalog. AWS Lake Formation permissions and registered lake locations will be used to receive temporary permissions.

The following example policy can be used in a role allowing a user of EMR Studio to use Athena workgroups to run queries.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/for-use-with-amazon-emr-managed-
policies": "true"
                }
            }
        },
        {
            "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/for-use-with-amazon-emr-managed-
policies": "true",
                    "ec2:CreateAction": "CreateSecurityGroup"
                }
            }
        },
        {
            "Sid": "AllowSecretManagerListSecrets",
```

```
            "Action": [
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect": "Allow",
            "Action": "secretsmanager:CreateSecret",
            "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/for-use-with-amazon-emr-managed-
policies": "true"
                }
            }
        },
        {
            "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect": "Allow",
            "Action": "secretsmanager:TagResource",
            "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/service-
role/AmazonEMRStudio_ServiceRole_Name"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "AllowS3ListAndLocationPermissions",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
```

```
            "Sid": "AllowS3ReadOnlyAccessToLogs",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-logs-Your-AWS-Account-ID-Region/
elasticmapreduce/*"
            ],
            "Effect": "Allow"
        },
        {
            "Sid": "AllowAthenaQueryExecutions",
            "Effect": "Allow",
            "Action": [
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:GetQueryResults",
                "athena:StopQueryExecution",
                "athena:ListQueryExecutions",
                "athena:GetQueryResultsStream",
                "athena:ListWorkGroups",
                "athena:GetWorkGroup",
                "athena:CreatePreparedStatement",
                "athena:GetPreparedStatement",
                "athena:DeletePreparedStatement"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGlueSchemaManipulations",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetPartition",
                "glue:GetPartitions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowQueryEditorToAccessWorkGroup",
            "Effect": "Allow",
```

```
            "Action": "athena:GetWorkGroup",
            "Resource": "arn:aws:athena:*:111122223333:workgroup*"
        },
        {
            "Sid": "AllowConfigurationForWorkspaceCollaboration",
            "Action": [
                "elasticmapreduce:UpdateEditor",
                "elasticmapreduce:PutWorkspaceAccess",
                "elasticmapreduce:DeleteWorkspaceAccess",
                "elasticmapreduce:ListWorkspaceAccessIdentities"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/creatorUserId":
"${aws:userId}"
                }
            }
        },
        {
            "Sid": "DescribeNetwork",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListIAMRoles",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AssumeRole",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

The following trust policy allows EMR Studio to assume the role:

> **Note**
>
> Additional permissions are needed to leverage EMR Studio Workspaces and EMR Notebooks. See Create permissions policies for EMR Studio users for more information.

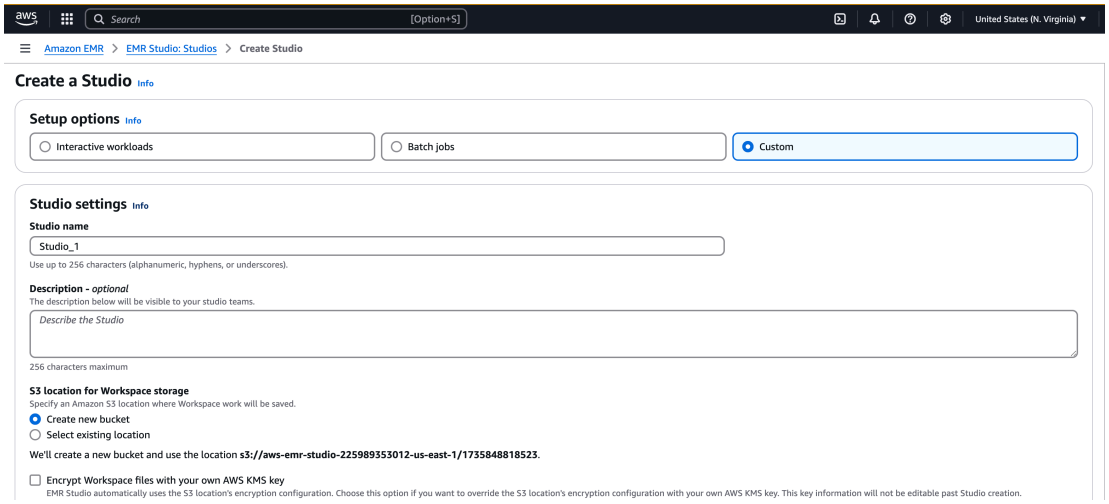
You can find more information with the following links:

- Define custom IAM permissions with customer managed policies
- EMR Studio service role permissions

Step 2. Create and configure your EMR Studio

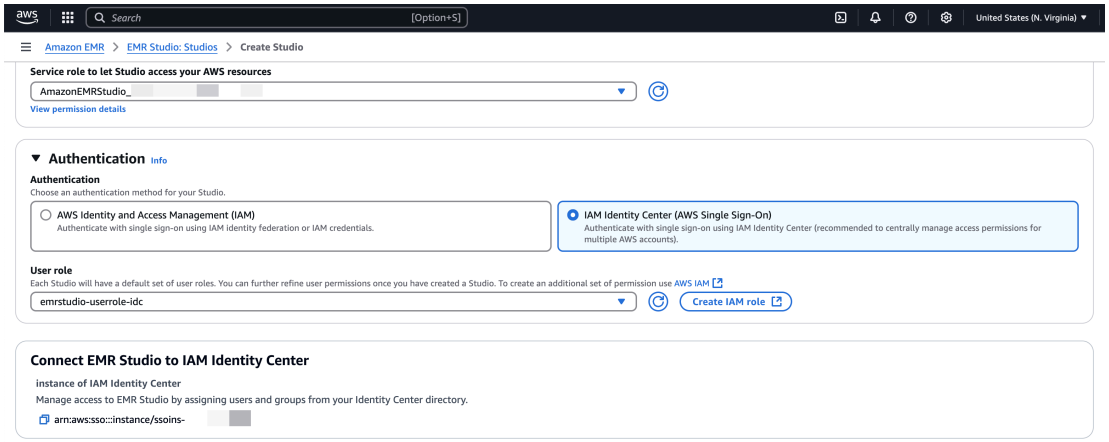
In this step, you'll create an Amazon EMR Studio in the EMR Studio console and use the IAM roles you created in Step 1. Create the required IAM roles for EMR Studio.

1. Navigate to the EMR Studio console, select **Create Studio** and the **Custom Setup** option. You can either create a new S3 bucket or use an existing bucket. You may check the box to **Encrypt workspace files with your own KMS keys**. For more information, see AWS Key Management Service.



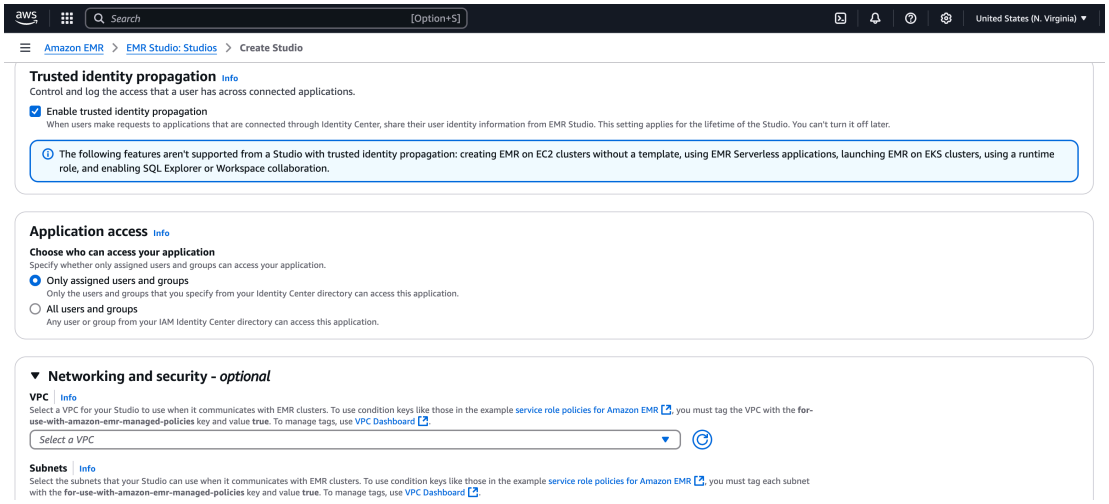
2. Under **Service role to let Studio access your resources**, select the service role created in Step 1. Create the required IAM roles for EMR Studio from the menu.

3. Choose **IAM Identity Center** under **Authentication**. Select the user role created in Step 1. Create the required IAM roles for EMR Studio.



4. Check the **Trusted identity propagation** box. Choose **Only assigned users and groups** under the Application access section, which will allow you to grant only authorized user and groups to access this studio.

5. *(Optional)* - You can configure VPC and subnet if you are using this Studio with EMR clusters.

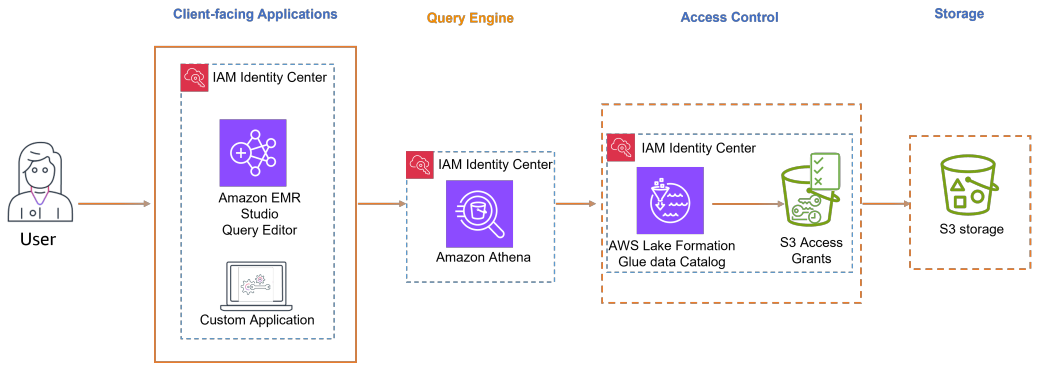


6. Review all the details and select **Create Studio**.
7. After configuring an Athena WorkGroup or EMR clusters, sign in to the Studio's URL to:
   a. Run Athena queries with the Query Editor.
   b. Run Spark jobs in the workspace using Jupyter notebook.

Trusted identity propagation with Amazon Athena

The steps to enable trusted identity propagation depend on whether your users interact with AWS managed applications or customer managed applications. The following diagram shows a trusted identity propagation configuration for client-facing applications - either AWS managed or external to AWS - that uses Amazon Athena to query Amazon S3 data with access control provided by AWS Lake Formation and Amazon S3 Access Grants.

> **Note**
>
> - Trusted identity propagation with Amazon Athena requires the use of Trino.
> - Apache Spark and SQL clients connected to Amazon Athena via ODBC and JDBC drivers are not supported.



AWS managed applications

The following AWS managed client-facing application supports trusted identity propagation with Athena:

- Amazon EMR Studio

To enable trusted identity propagation, follow these steps:

- Set up Amazon EMR Studio as the client-facing application for Athena. The Query Editor in EMR Studio is needed to run Athena Queries when trusted identity propagation is enabled.
- Set up Athena Workgroup.
- Set up AWS Lake Formation to enable fine-grained access control for AWS Glue tables based on the user or group in IAM Identity Center.
- Set up Amazon S3 Access Grants to enable temporary access to the underlying data locations in S3.

> **Note**
>
> Both Lake Formation and Amazon S3 Access Grants are required for access control to AWS Glue Data Catalog and for Athena query results in Amazon S3.

Customer managed applications

To enable trusted identity propagation for users of *custom-developed applications*, see to Access AWS services programmatically using trusted identity propagation in the *AWS Security Blog*.

Setting up trusted identity propagation with Amazon Athena workgroups

The following procedure walks you through setting up Amazon Athena workgroups for trusted identity propagation.

Prerequisites

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

1. Enable IAM Identity Center. Organization instance is recommended. For more information, see Prerequisites and considerations.
2. Provision the users and groups from your source of identities into IAM Identity Center.
3. This configuration requires Amazon EMR Studio, AWS Lake Formation, and Amazon S3 Access Grants.

Setting up trusted identity propagation with Athena

To set up trusted identity propagation with Athena, the Athena administrator must:

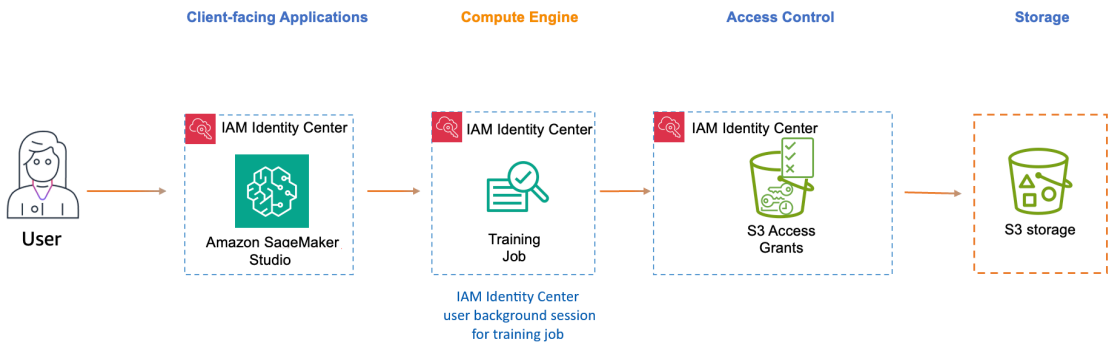
1. Review Considerations and limitations in using IAM Identity Center enabled Athena workgroups.
2. Create an IAM Identity Center enabled Athena workgroup.

Trusted identity propagation with Amazon SageMaker Studio

Amazon SageMaker Studio integrates with IAM Identity Center, and it supports user background sessions and trusted identity propagation. User background sessions allow a user to initiate a long-running job on SageMaker Studio, without that user having to remain signed in while the job runs. The job runs immediately and in the background, using the permissions of the user who initiated the job. The job can continue to run even if the user turns off their computer, their IAM Identity Center sign-in session expires, or the user signs out of the AWS access portal. The default session duration for user background sessions is 7 days, but you can specify a maximum duration of 90 days. Trusted identity propagation allows fine-grained access to be provided to AWS resources such as Amazon S3 buckets based on the user's identity or group membership.

The following diagram shows a trusted identity propagation configuration for SageMaker Studio, with access to data stored in an Amazon S3 bucket. User background sessions are enabled for IAM

Identity Center, which allows the SageMaker Studio training job to run in the background. Access control for the training data is provided by Amazon S3 Access Grants.



AWS managed application

The following AWS managed client-facing application supports trusted identity propagation:

- Amazon SageMaker Studio

To enable trusted identity propagation and user background sessions, follow these steps:

- Set up SageMaker Studio as the client-facing application.
- Set up Amazon S3 Access Grants to enable temporary access to the underlying data locations in Amazon S3.

Setting up trusted identity propagation with SageMaker Studio

The following procedure walks you through setting up SageMaker Studio for trusted identity propagation and user background sessions.

Prerequisites

Before you can get started with this tutorial, you'll need to complete the following tasks:

1. Enable IAM Identity Center. An organization instance is required. For more information, see Prerequisites and considerations.
2. Provision the users and groups from your source of identities into IAM Identity Center.
3. Confirm that user background sessions are enabled in the IAM Identity Center console. By default, user background sessions are enabled and the session duration is set to 7 days. You can change this duration.

To set up trusted identity propagation from SageMaker Studio, the SageMaker Studio administrator must perform the following steps.

Step 1: Enable trusted identity propagation in a new or existing SageMaker Studio domain

SageMaker Studio uses domains to organize user profiles, applications, and their associated resources. To enable trusted identity propagation, you must create a SageMaker Studio domain or modify an existing domain as described in the following procedure.

1. Open the SageMaker AI console, navigate to **Domains**, and do either of the following.
   - **Create a new SageMaker Studio domain by using Setup for organizations.**

     Choose **Set up for organizations**, and then do the following:
     - Choose **AWS Identity Center** as the authentication method.
     - Select the **Enable trusted identity propagation for all users on this domain** check box.
   - **Modify an existing SageMaker Studio domain.**
     - Select an existing domain that uses IAM Identity Center for authentication.

       > **Important**
       >
       > Trusted identity propagation is only supported in SageMaker Studio domains that use IAM Identity Center for authentication. If the domain uses IAM for authentication, you can't change the authentication method, and therefore you can't enable trusted identity propagation.

     - Edit domain settings. Edit the **Authentication and permissions** settings to enable trusted identity propagation.
2. Proceed to Step 2: Configure the default domain execution role. This role is required for users of a SageMaker Studio domain to access other AWS services such as Amazon S3.

Step 2: Configure the default domain execution role and role trust policy

A *domain execution role* is an IAM role that a SageMaker Studio domain assumes on behalf of all users in the domain. The permissions that you assign to this role determine what actions SageMaker Studio can perform.

1. To create or select a domain execution role, do either of the following:

- **Create or select a role by using Setup for organizations.**
  - Open the SageMaker AI console and follow the console guidance in **Step 2: Configure roles and ML activities** to create a new domain execution role or select an existing role.
  - Complete the rest of the setup steps to create your SageMaker Studio domain.
- **Create an execution role manually.**
  - Open the IAM console and create the execution role yourself.

2. Update the trust policy that is attached to the domain execution role so that it includes the following two actions: `sts:AssumeRole` and `sts:SetContext`. For information about how to find the execution role for your SageMaker Studio domain, see Get domain execution role.

   A *trust policy* specifies the identity that can assume a role. This policy is required to allow the SageMaker Studio service to assume the domain execution role. Add these two actions so that they appear as follows in your policy.

```
{

    "Version": "2012-10-17",


    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ]
        }
    ]
}
```

Step 3: Verify required Amazon S3 Access Grant permissions for the domain execution role

To use Amazon S3 Access Grants, you must have a permissions policy attached (either as an inline policy or a customer managed policy) to your SageMaker Studio domain execution role that contains the following permissions.

```
{

    "Version": "2012-10-17",

    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetDataAccess",
                "s3:GetAccessGrantsInstanceForPrefix"
                ],
            "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
        }
    ]
}
```

If you don't have a policy that contains these permissions, follow the instructions in Adding and removing IAM identity permissions in the *AWS Identity and Access Management User Guide*.

Step 4: Assign groups and users to the domain

Assign groups and users to the SageMaker Studio domain by following the steps in Add groups and users.

Step 5: Set up Amazon S3 Access Grants

To set up Amazon S3 Access Grants, follow the steps in Configuring Amazon S3 Access Grants for trusted identity propagation through IAM Identity Center. Use the step-by-step instructions to complete the following tasks:

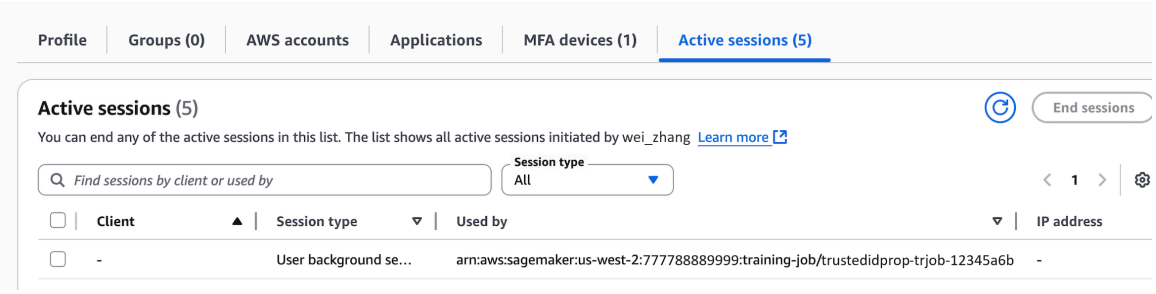
1. Create an Amazon S3 Access Grants instance.
2. Register a location in that instance.
3. Create grants to allow specific IAM Identity Center users or groups to access designated Amazon S3 locations or subsets (for example, specific prefixes) within those locations.

Step 6: Submit a SageMaker training job and view user background session details

In SageMaker Studio, launch a new Jupyter notebook and submit a training job. While the job is running, complete the following steps to view the session information and to verify that the user background session context is active.

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 active sessions for this user.
5. To search for sessions by the Amazon Resource Name (ARN) of the job that is using the session, in the **Session type** list, choose **User background sessions**, and then enter the job ARN in the search box.

Following is an example of how a training job that is using a user background session appears in the A**ctive sessions** tab for a user.



Step 7: View the CloudTrail logs to verify trusted identity propagation in CloudTrail

When trusted identity propagation is enabled, actions appear in CloudTrail event logs under the `onBehalfOf` element. The `userId` reflects the ID of the IAM Identity Center user who initiated the training job. The following CloudTrail event captures the process of trusted identity propagation.

```
                            "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA123456789EXAMPLE:SageMaker",
    "arn": "arn:aws:sts::111122223333:assumed-role/SageMaker-
ExecutionRole-20250728T125817/SageMaker",
```

```
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
        "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA123456789EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/service-role/SageMaker-
ExecutionRole-20250728T125817",
            "accountId": "111122223333",
            "userName": "SageMaker-ExecutionRole-20250728T125817"
        },
        "attributes": {
            "creationDate": "2025-07-29T17:17:10Z",
            "mfaAuthenticated": "false"
        }
    },
    "onBehalfOf": {
        "userId": "2801d3e0-f0e1-707f-54e8-f558b19f0a10",
        "identityStoreArn": "arn:aws:identitystore::777788889999:identitystore/
d-1234567890"
    }
},
```

Runtime considerations

If an administrator sets **MaxRuntimeInSeconds** for long-running training or processing jobs that is lower than the user background session duration, SageMaker Studio runs the job for the minimum of either **MaxRuntimeInSeconds** or the user background session duration.

For more information about **MaxRuntimeInSeconds**, see the guidance for the `CreateTrainingJob` StoppingCondition parameter in the *Amazon SageMaker API Reference*.

Authorization services

In all analytics and data lakehouse use cases, you can achieve fine-grained access controls using:

- AWS Lake Formation - for guidance, see Setting up AWS Lake Formation with IAM Identity Center.
- Amazon S3 Access Grants - for guidance, see Setting up Amazon S3 Access Grants with IAM Identity Center.

Setting up AWS Lake Formation with IAM Identity Center

AWS Lake Formation is a managed service that simplifies the creation and management of data lakes on AWS. It automates data collection, cataloging, and security, providing a centralized repository for storing and analyzing diverse data types. Lake Formation offers fine-grained access controls and integrates with various AWS analytics services, enabling organizations to efficiently set up, secure, and derive insights from their data lakes.

Follow these steps to enable Lake Formation to grant data permissions based on user identity using IAM Identity Center and trusted identity propagation.

Prerequisites

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

- Enable IAM Identity Center. Organization instance is recommended. For more information, see Prerequisites and considerations.

Steps to set up trusted identity propagation

1. **Integrate IAM Identity Center with AWS Lake Formation** following the guidance in Connecting Lake Formation with IAM Identity Center.

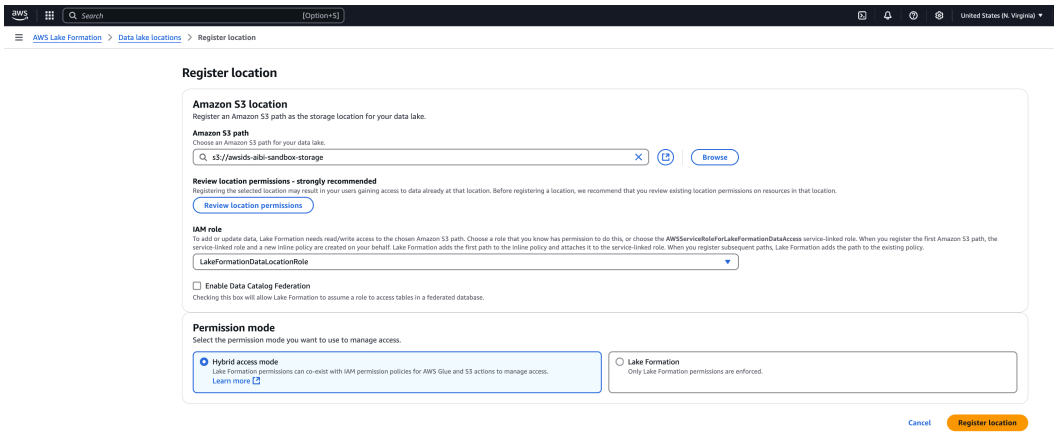
   > **Important**
   >
   > **If you do not have AWS Glue Data Catalog tables**, you must create them in order to use AWS Lake Formation to grant access to IAM Identity Center users and groups. See Creating objects in AWS Glue Data Catalog for more information.

2. **Register data lake locations**.

   Register the S3 locations where the data of the Glue tables are stored. By doing this, Lake Formation will provision temporary access to the required S3 locations when the tables are queried, removing the need to include S3 permissions in the service role (e.g. the Athena service role configured on the WorkGroup).

   a. Navigate to the **Data lake locations** under the **Administration** section in the navigation pane in the AWS Lake Formation console. Select **Register location**.

This will allow Lake Formation to provision temporary IAM credentials with the necessary permissions to access S3 data locations.



b. Enter the S3 path of the data locations of the AWS Glue tables in the **Amazon S3 path** field.

c. In the **IAM role** section, do not select the service linked role if you want to use it with trusted identity propagation. Create a separate role with the following permissions.

To use these policies, replace the *italicized placeholder text* in the example policy with your own information. For additional directions, see Create a policy or Edit a policy. The permission policy should grant access to the S3 location specified in the path:

i. **Permission policy**:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LakeFormationDataAccessPermissionsForS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::Your-S3-Bucket/*"
```

```
            ]
        },
        {
            "Sid":
 "LakeFormationDataAccessPermissionsForS3ListBucket",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::Your-S3-Bucket"
            ]
        },
        {
            "Sid": "LakeFormationDataAccessServiceRolePolicy",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        }
    ]
}
```

ii. **Trust relationship**: This should include `sts:SectContext`, which is required for trusted identity propagation.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "lakeformation.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
```

```
            ]
        }
    ]
}
```

> **Note**
>
> The IAM role created by the wizard is a service-linked role and does not include `sts:SetContext`.

d. After creating the IAM role, select **Register location**.

Trusted identity propagation with Lake Formation across AWS accounts

AWS Lake Formation supports using AWS Resource Access Manager (RAM) to share tables across AWS accounts and it works with trusted identity propagation when the grantor account and grantee account are in the same AWS Region, in the same AWS Organizations, and share the same organization instance of IAM Identity Center. See Cross-account data sharing in Lake Formation for more information.

Setting up Amazon S3 Access Grants with IAM Identity Center

Amazon S3 Access Grants provides the flexibility to grant identity-based fine-grain access control to S3 locations. You can use Amazon S3 Access Grants to grant Amazon S3 bucket access directly to your corporate users and groups. Follow these steps to enable S3 Access Grants with IAM Identity Center and achieve trusted identity propagation.

Prerequisites

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

- Enable IAM Identity Center. Organization instance is recommended. For more information, see Prerequisites and considerations.

Configuring S3 Access Grants for trusted identity propagation through IAM Identity Center

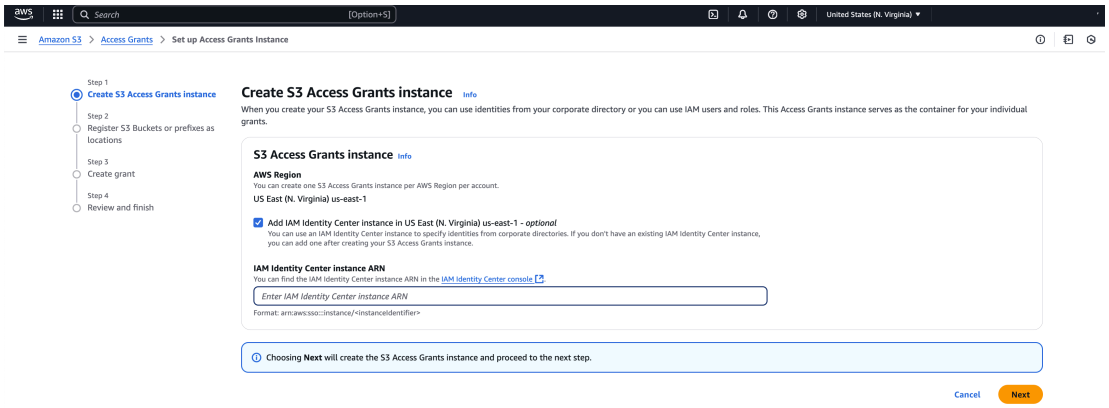
If you already have an Amazon S3 Access Grants instance with a registered location, follow these steps:

1. Associate your IAM Identity Center instance.
2. Create a grant.

If you have not created an Amazon S3 Access Grants yet, follow these steps:

1. **Create an S3 Access Grants instance** - You can create one S3 Access Grants instance per AWS Region. When you create the S3 Access Grants instance, make sure to check the **Add IAM Identity Center instance** box and provide the ARN of your IAM Identity Center instance. Select **Next**.

   The following image shows the Create S3 Access Grants instance page in the Amazon S3 Access Grants console:



2. **Register a location** - After you create an create an Amazon S3 Access Grants instance in an AWS Region in your account, you register an S3 location in that instance. An S3 Access Grants location maps the default S3 region (`S3://`), a bucket, or a prefix to an IAM role. S3 Access Grants assumes this Amazon S3 role to vend temporary credentials to the grantee that is accessing that particular location. You must first register at least one location in your S3 Access Grants instance before you can create an access grant.

   For the **Location scope**, specify `s3://`, which includes all of your buckets in that Region. This is the recommended location scope for most use cases. If you have an advanced access management use case, you can set the location scope to a specific bucket `s3://`*`bucket`*or

prefix within a bucket `s3://bucket/prefix-with-path`. For more information, see Register a location in the *Amazon Simple Storage Service User Guide*.

> **Note**
>
> Ensure the S3 locations of the AWS Glue tables you want to grant access to are included in this path.

The procedure requires you to configure an IAM role for the location. This role should include permissions to access the location scope. You can use the S3 console wizard to create the role. You'll need to specify your S3 Access Grants instance ARN in the policies for this IAM role. The default value of your S3 Access Grants instance ARN is `arn:aws:s3:Your-Region:Your-AWS-Account-ID:access-grants/default`.

The following example permission policy gives Amazon S3 permissions to the IAM role that you created. And the example trust policy following it allows the S3 Access Grants service principal to assume the IAM role.

a. **Permission policy**

   To use these policies, replace the *italicized placeholder text* in the example policy with your own information. For additional directions, see Create a policy or Edit a policy.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ObjectLevelReadPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetObjectAcl",
                "s3:GetObjectVersionAcl",
                "s3:ListMultipartUploadParts"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                },
                "ArnEquals": {
                    "s3:AccessGrantsInstanceArn": [
                    "arn:aws:s3:::access-grants/instance-id"
                    ]
                }
            }
        },
        {
            "Sid": "ObjectLevelWritePermissions",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectVersionAcl",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:AbortMultipartUpload"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ],
            "Condition": {
                "StringEquals": {
                "aws:ResourceAccount": "111122223333"
                },
                "ArnEquals": {
                    "s3:AccessGrantsInstanceArn": [
                    "arn:aws:s3:::access-grants/instance-id"
                    ]
                }
            }
        },
        {
            "Sid": "BucketLevelReadPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ],
            "Condition": {
                "StringEquals": {
                "aws:ResourceAccount": "111122223333"
                },
                "ArnEquals": {
                    "s3:AccessGrantsInstanceArn": [
                    "arn:aws:s3:::access-grants/instance-id"
                    ]
                }
            }
        },
        {
            "Sid": "OptionalKMSPermissionsForSSEEncryption",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

b. **Trust policy**

In the IAM role trust policy, give the S3 Access Grants service (`access-grants.s3.amazonaws.com`) principal access to the IAM role that you created. To do so, you can create a JSON file that contains the following statements. To add the trust policy to your account, see Create a role using custom trust policies.

JSON

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

```
            "Sid": "Stmt1234567891011",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "Your-Custom-Access-Grants-Location-
ARN"
                }
            }
        },

        {
            "Sid": "Stmt1234567891012",
            "Effect": "Allow",
            "Action": "sts:SetContext",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "Your-Custom-Access-Grants-Location-
ARN"
                },
                "ForAllValues:ArnEquals": {
                    "sts:RequestContextProviders":
 "arn:aws:iam::aws:contextProvider/IdentityCenter"
                }
            }
        }
    ]
}
```

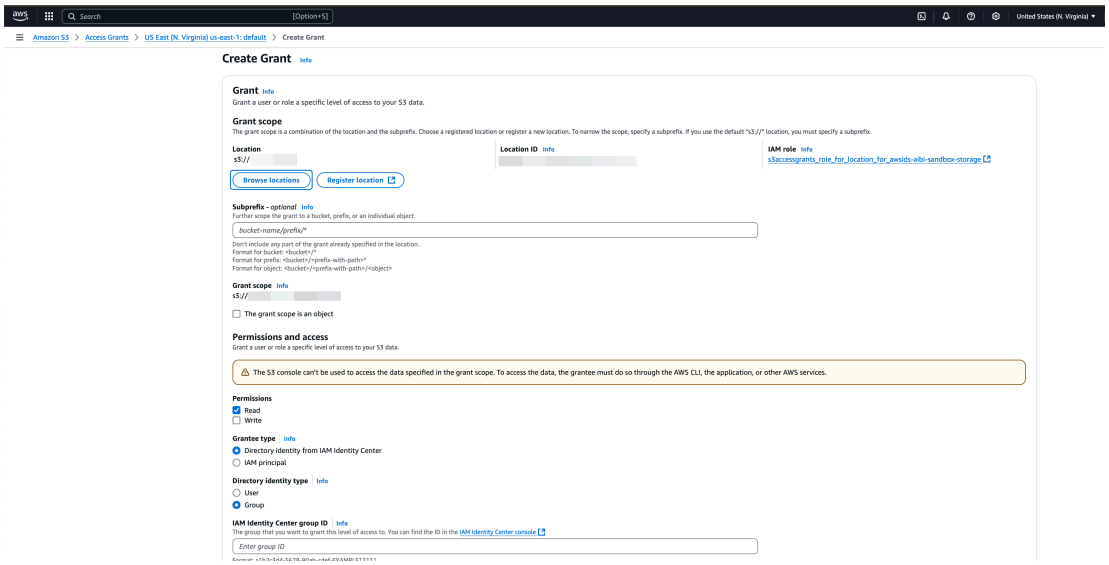
Create an Amazon S3 Access Grant

If you have an Amazon S3 Access Grants instance with a registered location and you have associated your IAM Identity Center instance with it, you can create a grant. In the S3 console **Create Grant** page, complete the following:

Create a grant

1. Select the location created in the previous step. You can reduce the scope of the grant by adding a sub-prefix. The sub-prefix can be a `bucket`, `bucket/prefix`, or an object in the bucket. For more information, see Subprefix in the *Amazon Simple Storage Service User Guide*.
2. Under **Permissions and access**, select **Read** and or **Write** depending on your needs.
3. In **Granter type**, choose **Directory Identity form IAM Identity Center**.
4. Provide the IAM Identity Center **User or Group ID**. You can find the user and group IDs in the IAM Identity Center console under **User** and **Group** sections. Select **Next**.
5. On the **Review and Finish** page, review the settings for the S3 Access Grant and then select **Create Grant**.

   The following image shows the Create Grant page in the Amazon S3 Access Grants console:



Setting up your own OAuth 2.0 application

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
- Using applications with a trusted token issuer

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 **Enable refresh token grant**. 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 `CreateTokenWithIAM` API action. For more information about this policy, and an example that you can adapt as required for your environment, see Resource-based policy example for IAM Identity Center IAM Identity Center.

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 are 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 do not 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 are 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
- Identity-enhanced IAM role sessions

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 cannot 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.

> **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.

  > **Note**
  >
  > Using a custom signing key for JWTs from Microsoft Entra ID is not supported. In order to use tokens from Microsoft Entra ID with trusted token issuer, you can't use a custom signing key.

- **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.

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 will not 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 443 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	Adds the external IdP as a trusted token issuer to the IAM Identity Center console. Helps set up the correct attribute mapping between IAM Identity Center and the external IdP. Notifies the AWS service administrator when the trusted token issuer is added to the IAM Identity Center console.	External IdP (trusted token issuer) administrator AWS service administrator
External IdP (trusted token issuer) administrator	Configures the external IdP to issue tokens. Helps set up the correct attribute mapping between IAM Identity Center and the external IdP. Provides the audience name (Aud claim) to the AWS service administrator.	IAM Identity Center administrator AWS service administrator
AWS service administrator	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 administr	IAM Identity Center administrator External IdP (trusted token issuer) administrator

Role	Performs these tasks	Coordinates with
	ator adds it to the IAM Identity Center console. Configures the AWS service to use the trusted token issuer.	

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.

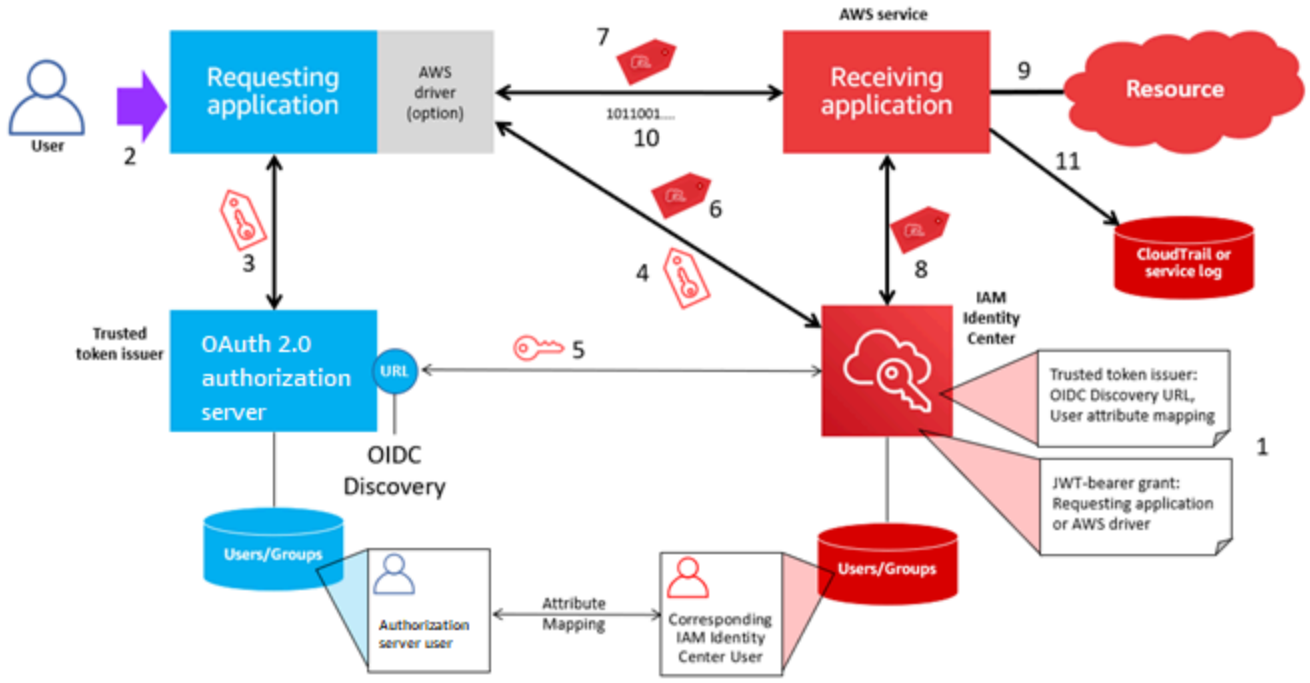
To view or edit trusted token issuer settings in the IAM Identity Center console

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 are 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 are recorded in the receiving application logs and CloudTrail events. The specific way in which this information is logged varies based on the application.

Identity-enhanced IAM role sessions

The AWS Security Token Service (STS) enables an application to obtain an identity-enhanced IAM role session. Identity-enhanced role sessions have an added identity context that carries a user identifier to the AWS service that it calls. AWS services can look up the group memberships and attributes of the user in IAM Identity Center and use them to authorize the user's access to resources.

AWS applications obtain identity-enhanced role sessions by making requests to the AWS STS AssumeRole API action and passing a context assertion with the user's identifier (`userId`) in the `ProvidedContexts` parameter of the request to `AssumeRole`. The context assertion is obtained from the `idToken` claim received in response to a request to `SSO OIDC` to `CreateTokenWithIAM`. When an AWS application uses an identity-enhanced role session to access a resource, CloudTrail logs the `userId`, the initiating session, and the action taken. For more information, see Identity-enhanced IAM role session logging.

Topics

- Types of identity-enhanced IAM role sessions
- 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. Applications that have obtained Id tokens from IAM Identity Center can add `sts:identiy_context` (recommended) or `sts:audit_context` (Supported for backward compatibility) to IAM role sessions. An identity-enhanced IAM role session can have only one of these context assertions, not both.

Identity-enhanced IAM role sessions created with `sts:identity_context`

When an identity-enhanced role session contains `sts:identity_context` the called AWS service determines if resource authorization is based on the user who is represented in the role session, or if it is based on the role. AWS services that support user-based authorization provide the application's administrator with controls to assign access to the user or to groups for which the user is a member.

AWS services that do not support user-based authorization disregard the `sts:identity_context`. CloudTrail logs the userId of the IAM Identity Center user with all actions taken by the role. For more information, see Identity-enhanced IAM role session logging.

To obtain this type of identity-enhanced role session from AWS STS, 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`.

For more information on how the authorization behaves, see the documentation for the receiving AWS service.

Identity-enhanced IAM role sessions created with `sts:audit_context`

In the past, `sts:audit_context` was used to enable AWS services to log the user identity without using it to make an authorization decision. AWS services are now able to use a single context - `sts:identity_context` - to achieve this as well as to make authorization decisions. We recommend using `sts:identity_context` in all new deployments of trusted identity propagation.

Identity-enhanced IAM role session logging

When a request is made to an AWS service using an identity-enhanced IAM role session, 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": "AROAEXAMPLE:MyRole",
      "arn": "arn:aws:sts::111111111111:assumed-role/MyRole/MySession",
      "accountId": "111111111111",
      "accessKeyId": "ASIAEXAMPLE",
      "sessionContext": {
        "sessionIssuer": {
            "type": "Role",
            "principalId": "AROAEXAMPLE",
            "arn": "arn:aws:iam::111111111111:role/MyRole",
            "accountId": "111111111111",
            "userName": "MyRole"
        },
        "attributes": {
            "creationDate": "2023-12-12T13:55:22Z",
            "mfaAuthenticated": "false"
        }
    },
    "onBehalfOf": {
        "userId": "11111111-1111-1111-1111-1111111111",
        "identityStoreArn": "arn:aws:identitystore::111111111111:identitystore/
d-111111111"
    }
}
```

Rotate 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*.

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 do not 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

In the IAM Identity Center console, the **Applications** page displays status indicator icons in the properties of each application. These icons display 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 displays 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 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.

Understand 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.

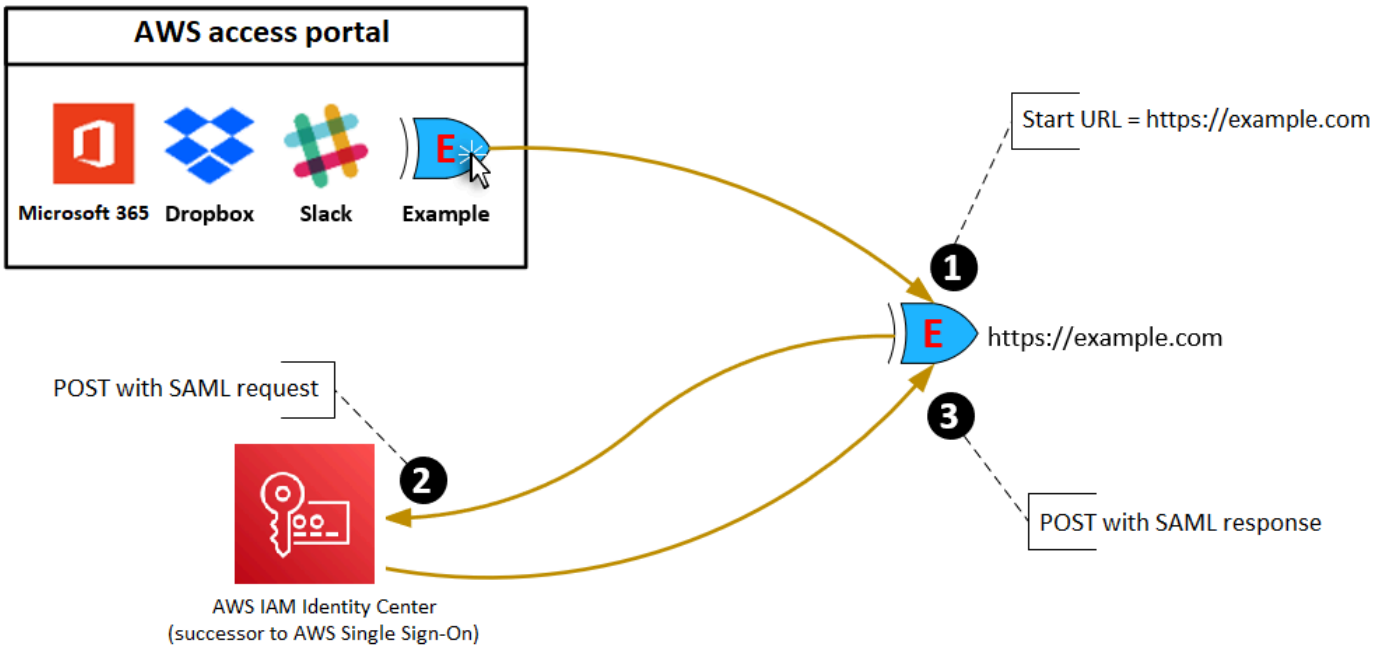
Application start URL

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:

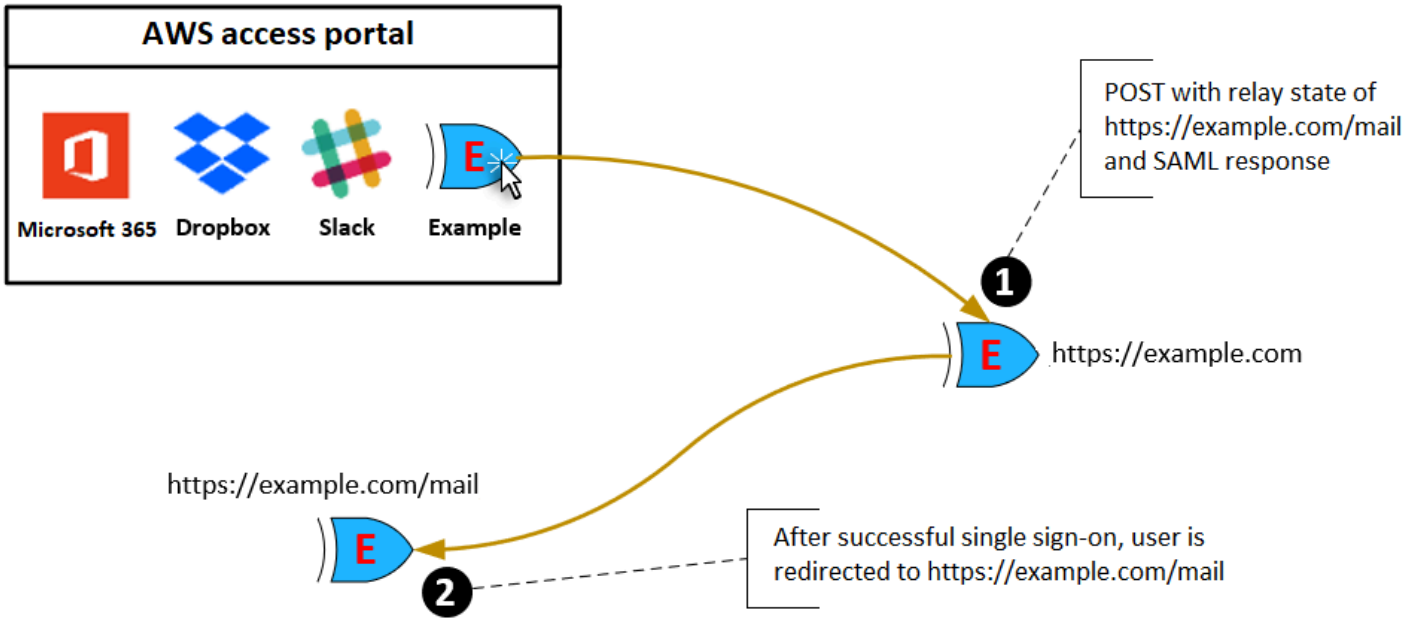
1. The user's browser redirects the authentication request using the value for the application start URL (in this case https://example.com).
2. The application sends an `HTML POST` with a `SAMLRequest` to IAM Identity Center.

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 display name or group name. 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 to SAML 2.0 applications

Use this procedure to remove user access to SAML 2.0 applications in the application catalog or custom SAML 2.0 applications. For more information on authentication sessions and durations, see Understanding authentication sessions in IAM Identity Center.

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 between IAM Identity Center and External Identity Providers directory.
8. In the third column of the table, choose the appropriate format for the attribute from the menu.
9. Choose **Save changes**.

Configure 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 do not 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
Add, edit, or delete users or groups	❌ No	✅ Yes*	✅ Yes

IAM Identity Center administrative tasks	Member account	Delegated administrator account	Management account
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
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

IAM Identity Center administrative tasks	Member account	Delegated administrator account	Management account
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

*Refer to the best practices for delegated administration regarding user and group assignments to the management account.

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 Configure 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 Setting up and 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 Understanding service-linked roles in IAM Identity Center 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**
>
> To ensure your AWS managed applications function well with your IAM Identity Center you should avoid applying this SCP to the AWS accounts where you deployed those applications. Also, if you use delegated administration, you should avoid applying this SCP to the delegated administration account. For more information, see Best practices.

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 Configure 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. You can also consider using temporary elevated access to grant this access only when needed.

- **Dedicated permission sets for the management account** – Use dedicated permission sets for the management account. For security reasons, a permission set used for access to the management account can only be modified by an IAM Identity Center administrator from the management account. The delegated administrator can't alter permission sets provisioned in the management account.
- **Assign users only (not groups) to permission sets in the management account** – Because the management account has special privileges, you must use caution when assigning access to this account in the console or AWS Command Line Interface (CLI). If you assign groups to permission sets with access to the management account, anyone with permissions to modify the memberships in those groups can add/remove users to/from those groups, and thus affect who has access to the management account. This is any group admin with control over your identity source, including your identity provider (IdP) administrator, Microsoft Active Directory Domain Service (AD DS) administrator, or IAM Identity Center administrator. Therefore, you should assign users directly to permission sets that grant access in the management account, and avoid groups. If you do use groups to manage access to the management account, ensure that proper controls are in place in the IdP to limit who has the ability to modify those groups, and ensure that changes to those groups (or changes to the credentials for the users in the management account) are logged and reviewed as necessary.
- **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 the IAM Identity Center delegated administrator member account. If you want your Active Directory to be in the management account, you must perform the setup in the management account as the delegated administrator won't have the necessary permissions to complete it.

Limit IAM Identity Center identity store actions in the delegated administration account with external identity sources

If you use an external identity source such as an IdP or Directory Service, you should implement policies that limit the identity store actions that an IAM Identity Center admin can take from within the delegated administration account. Write and delete operations should be carefully considered. Generally, the external identity source is the source of truth for users and their attributes, and for

group memberships. If you modify these using the identity store APIs or the console, your changes will be overwritten during normal synchronization cycles. It's best to leave these operations to the exclusive control of your identity source of truth. This also guards against an IAM Identity Center administrator modifying group memberships to grant access to a group-assigned permission set or application, rather than leaving the group membership control to your IdP admin. You should also guard who can create SCIM bearer tokens from the delegated administration account, as these could enable a member account admin to modify groups and users through a SCIM client.

There may be times when write or delete operations are appropriate from the delegated admin account. For example, you can create a group without adding members, then make assignments to a permission set without having to wait for the IdP admin to create the group. No one will have access to that assignment until the IdP admin provisions the group and the IdP sync process establishes the group members. It may also be appropriate to delete a user or a group to prevent sign-in or authorization during a time when you're unable to wait on the IdP sync process to remove access by the user or the group. However, misuse of this permission can be disruptive to users. You should use the principle of least privilege when assigning identity store permissions. You can control which identity store actions are allowed by your delegated administration account admins using a service control policy (SCP).

The example SCP below prevents assigning users to groups through the Identity Store API and the AWS Management Console, which is recommended when your identity source is external. This does not affect user sync from Directory Service or from an external IdP (via SCIM).

> **Note**
>
> It is possible that, although you use an external identity source, your organization relies, fully or partly, on the Identity Store APIs for the provisioning of users and groups. Therefore, before activating this SCP, you should confirm that your user provisioning process does not use this Identity Store API operation. Also, refer to the next section for information about how to limit the managing of group memberships to specific groups.

```
{
  "Version": "2012-10-17",
  "Statement": [
    { "Effect": "Deny",
      "Action": ["identitystore:CreateGroupMembership"],
      "Resource": [ "*" ] }
```

```
  ]
}
```

If you'd like to prevent adding users only to groups that grant access to the management account, you can reference those specific groups using the group ARN in the following format: `arn:${Partition}:identitystore:::group/${GroupId}`. This and other resource types available in the Identity Store are documented in Resource types defined by AWS Identity Store in the *Service Authorization Reference*. You can also consider including additional Identity Store APIs in the SCP. For more information, see Actionsin the Identity Store API Reference.

By adding the following policy statement to your SCP, you can prevent the creation of SCIM bearer tokens by the delegated admin. You can apply this for both external identity sources.

> **Note**
>
> If your delegated admin needs to set up user provisioning with SCIM, or perform the periodic SCIM bearer token rotation, you will need to temporarily allow access to this API to allow the delegated admin to complete those tasks.

```
{ "Effect": "Deny",
  "Action": ["sso-directory:CreateBearerToken"],
  "Resource": [ "*" ]
}
```

Limit IAM Identity Center identity store actions in the delegated administration account for locally managed users

If you create your users and groups directly in IAM Identity Center, rather than using an external IdP or Directory Service, then you should take precautions for who can create users, reset passwords, and control group membership. These actions give the administrator great powers for who can sign in and who can gain access through membership in groups. These policies are best implemented as in-line policies within the permission sets you use for your IAM Identity Center administrators, rather than as SCPs. The following example inline policy has two objectives. Firstly, it prevents adding users to specific groups. You can use this to prevent delegated admins from adding users to groups that grant access to the management account. Secondly, it prevents the issuance of SCIM bearer tokens.

```
{
  "Version": "2012-10-17",
  "Statement": [
  { "Effect": "Deny",
    "Action": ["identitystore:CreateGroupMembership"],
    "Resource": [ arn:${Partition}:identitystore:::group/${GroupId1},
                  arn:${Partition}:identitystore:::group/${GroupId2}
                 ]
   }
  ],
  { "Effect": "Deny",
    "Action": ["sso-directory:CreateBearerToken"],
    "Resource": [ "*" ] }
  ]
}
```

Segregate IAM Identity Center configuration management from PermissionSet management

Separate the administrative tasks including modification of external identity source, SCIM token management, session timeout configuration from the tasks to create, modify, and assign permission sets by creating distinct admin permission sets from your management account.

Limit issuance of SCIM bearer tokens

SCIM bearer tokens enable an external identity source to provision users, groups, and group memberships via the SCIM protocol when the identity source of your IAM Identity Center is an external IdP such as Okta or Entra ID. You can set up the following SCP to prevent the creation of SCIM bearer tokens by delegated administrators. If your delegated administrator needs to set up user provisioning with SCIM, or perform the periodic SCIM bearer token rotation, you will need to temporarily allow access to this API to allow the delegated administrator to complete those tasks.

```
    { "Effect": "Deny",
      "Action": ["sso-directory:CreateBearerToken"],
      "Resource": [ "*" ]
}
```

Use permission set tags and account lists to delegate administration of specific accounts

You can create permissions sets that you assign to your IAM Identity Center administrators to delegate who can create permission sets, and who can assign which permission sets in which accounts. This is done by tagging permission sets and using policy conditions in permission sets that you assign to your administrators. For example, you can create permission sets that enable a user to create permission sets providing they are tagged a certain way. You can also create policies that enable an administrator to assign permission sets that have a specific tag in specified accounts. This can help you delegate management over accounts without giving an administrator the privileges to modify their access and privileges over the delegated administration account. For example, by tagging permission sets that you use only in the delegated administration account, you can specify a policy that gives only certain people the permissions to modify permission sets and assignments that affect the delegated administration account. You can also give other people permissions to manage a list of accounts outside of the delegated administration account. To learn more, see Delegating permission set management and account assignment in AWS IAM Identity Center in the *AWS Security Blog*.

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 is 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 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.

> **Note**
>
> To ensure business continuity, we recommend that you set up emergency access to the AWS Management Console.

To address a range of customers' needs, AWS IAM Identity Center integrates with the solutions from AWS Security Competency partners. AWS validates that these solutions 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 unique needs and preferences, including your business, the architecture of your cloud environment, and your budget.

Validated solutions include Apono Access Management Platform, CyberArk Secure Cloud Access, Okta Access Requests, and Tenable (previously Ermetic).

Partners can nominate solutions using the AWS Security Competency application in Partner Center. For more information, see AWS Security Competency Partners.

> **Note**
>
> If you are using resource-based, Amazon Elastic Kubernetes Service or AWS Key Management Service, 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.

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 or group 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 or group 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 and change a permission set.

> **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 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 .

   > **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**.

Considerations

- 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.

8. 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 are not using RADIUS MFA with AWS Directory Service.

   Note

   If you are 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.

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.

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. Alternatively, you can use the delete-account-assignment AWS CLI.

> **Note**
>
> When you need to deprovision IAM Identity Center users or groups, you should first remove any assignments of permission sets from your users and groups before deleting the users and groups.

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 are using the IAM Identity Center directory as an identity source, do not 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**.

   After deleting a user session, the user will immediately lose access to the AWS access portal. 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 will not 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.

Alternatively, you can use AWS CLI to create, attach policies to, and assign permission sets. The following lists the commands for each step:

- To create a permission set: create-permission-set
- To attach AWS Managed Policy to a permission set: attach-managed-policy-to-permission-set
- To attach customer managed policy to a permission set: attach-customer-managed-policy-to-permission-set
- To assign a permission set to a principal: create-account-assignment

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 **AWSSSOMasterAccountAdministrator** 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 or group 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.

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 are 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 in IAM Identity Center.

- **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 user or group access to AWS accounts.

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 do not 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.

> **Note**
>
> When a customer managed policy is attached to a permission set, the name of the policy is not case sensitive.

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 cannot 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.

> **Note**
>
> You can search and sort permission sets by name in the IAM Identity Center console.

Keep the following considerations in mind when creating permissions sets:

- **Organization instance**

  To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

- **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 are signed out of the console and asked to sign in again. As a security best practice, we recommend that you do not 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 in IAM Identity Center.

- **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
- Remove permission sets in IAM Identity Center
- Delete permission sets in IAM Identity Center

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.

> **Note**
>
> To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

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 do not 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.

> **Note**
>
> Permission set names must be unique within your IAM Identity Center instance.

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 Assign user or group access to AWS accounts.

View and change a permission set

You can use permission sets to grant users access to AWS accounts. You can view and change a permission set with the AWS IAM Identity Center console. You can search and sort permission sets by name in the 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”.

Permission sets are not required to manage user access to applications.

> **Note**
>
> To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

View permission set assignments

Use this procedure to view applied permission set in the AWS IAM Identity Center console.

All AWS accounts where a permission set is provisioned

To view all the assignments for a permission set, use the following procedure:

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. Once on the selected permission sets page, under the **Accounts** tab, you can see the accounts where the permission set is used. You can select the account to see how the permission set is provisioned within the account. You can delete, edit, and attach policies to the permission set.

All permission sets for an AWS account

To view all the assignments for a permission set, use the following procedure:

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**. Select the account for which you want to view the provisioned permission sets.
3. Once on the selected AWS account page, under the **Permission sets** tab, you can view the different permission set assigned to the selected AWS account. You can select the permission set hyperlink to learn more about the permission set.

All applied permission sets to users and groups

To view all the permission sets assigned to users or groups, use the following procedure:

1. Sign in to the AWS Management Console and open the AWS IAM Identity Center console at https://console.aws.amazon.com/singlesignon/.
2. Select either Users or Groups under **Dashboard** to view IAM Identity Center users or groups.
   a. Once on the **Users** page, select the user for whom you want to see applied permission sets. Next, select the **AWS accounts** tab and the AWS account under the **AWS account access** section. You'll be able to see the applied permission sets and AWS account for the selected user.
   b. Once on the **Groups** page, select the group you want to view applied permission sets. Next, select the **AWS accounts** tab and the AWS account under the **AWS account access** section. You'll be able to see the applied permission sets and AWS account for the selected group.

Change a permission set

Use this procedure to change a permission set with the IAM Identity Center console. You can add or remove permission sets from users or groups.

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.

> **Note**
>
> To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

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}`

Resource Types	ARN	Context Keys
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.

Considerations

- To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.
- 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.

> **Note**
>
> When a customer managed policy is attached to a permission set, the name of the policy is not case sensitive.

- The permissions that the policy grants do not 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.

Remove permission sets in IAM Identity Center

You can remove a permission set from IAM Identity Center users and groups in the IAM Identity Center console. You can also remove a permission set from an AWS account. For more information about permission sets and how they are used in IAM Identity Center, see Manage AWS accounts with permission sets.

> **Note**
>
> To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.

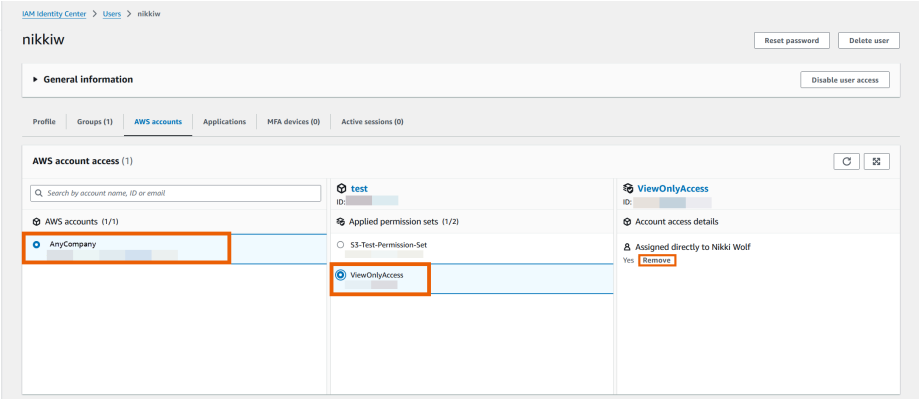
Remove permission set from a user

Remove permission set from a user

Use this procedure to remove a permission set from a user with the 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 **IAM Identity Center**, select **Users**.
3. Select the username of the user you want to remove a permission set from.
4. On the user details page, select the **AWS accounts** tab. Under **AWS account access**, select your AWS account.

5. In the right pane, the applied permissions for the selected user appears. Select the permission set you want to remove. Under **Account Access details**, select **Remove**.
6. A dialog box appears asking if you want to remove this permission set. Select **Remove**.

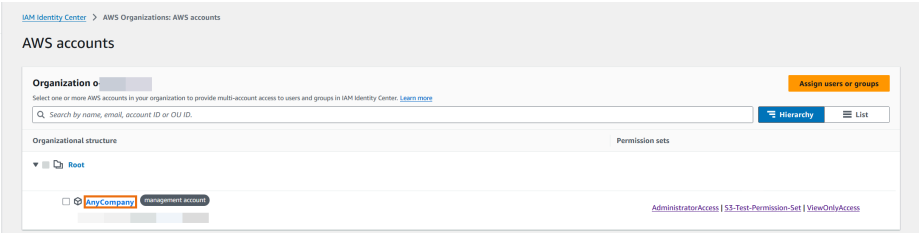


Remove permission set from a group

Remove permission set from a group

Use this procedure to remove a permission set from a group with the 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**, select **AWS accounts**. Select the link to your management account.



3. Under the **Assigned users and groups** tab, select the group you want to remove the permission set from and then select **Change permission set**.
4. On the **Change permission sets** page, clear the permission set you want to remove and then select **Save changes**.

Remove permission set from an AWS account

Use this procedure to remove a permission set from the AWS account with the 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**, select **AWS accounts**. Select the name of the AWS account from which you want to remove the permission set.
3. On the **Overview** page of the AWS account, choose the **Permission sets** tab. Select the permission set you want to remove. Then select **Remove**.
4. 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**.

Delete permission sets in IAM Identity Center

Before you can delete a permission set from IAM Identity Center, you should remove it from all AWS accounts that use the permission set. To check existing user and group access, see View and change a permission set.

Considerations

- To use permission sets, you'll need to use an Organization instance of IAM Identity Center. For more information, see Organization and account instances of IAM Identity Center.
- If you want to revoke an active permission set session, see the section called “End active sessions for workforce users”.
- You should remove permission sets and applications assignments from users or groups you want to delete before deleting them. Otherwise, you'll have unassigned and unused permission sets and applications in IAM Identity Center.

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.

> **Important**
>
> 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 and change a permission set.

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, administrators can complete the following configuration and management tasks to control user access and session duration.

Task	Learn more
Administrators can set the maximum duration for user sessions when accessing AWS resources through IAM Identity Center.	Set session duration for AWS accounts
Administrators can customize the landing page users see after successfully authenticating through IAM Identity Center.	Set relay state for quick access to the AWS Management Console
Ensure users no longer have access to AWS resources when their permissions are revoked.	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 is 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.

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 are 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 the section called "End active sessions for workforce users" 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.

JSON

```
{
```

```
    "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 is a globally unique value that is 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
`AWSReservedSSO_` *`permission-set-name_unique-suffix*	`arn:aws:iam::` *`aws-account-ID`*`:role/aws-reserved/sso.amazonaws.com/` *`aws-region`* `/AWSReser`

Name	ARN
	vedSSO_ *permission-set-name_unique-suffix*

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_ *permission-set-name_unique-suffix*	arn:aws:iam:: *aws-account-ID* :role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_ *permission-set-name_unique-suffix*

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_DatabaseAdministrator_` **`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 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 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 is 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 is 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 is 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.

JSON

```
{
  "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 do not 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.	• 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.	• 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.	• Getting started
	• 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.	• Choosing attributes when using an external identity provider as your identity source

Step	Task	Reference
	• If you choose to have your IdP send attributes, configure your IdP to transmit the attributes in SAML assertions. See the `Optional` sections in the tutorial for your specific IdP.	• IAM Identity Center identity source tutorials
	• If you use an IdP as your identity source and choose to select attributes in IAM Identity Center, investiga te 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.	• Provision users and groups from an external identity provider using SCIM • Supported external identity provider attributes
	• If you use Active Directory or IAM Identity Center as your identity source, or you use an IdP and choose to select attributes in IAM Identity Center, review the available attributes that you can configure. Then jump immediately to step 4 to start configuring your ABAC attributes using the IAM Identity Center console.	• 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 between IAM Identity Center and Microsoft AD
4	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.	• 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 `aws:PrincipalTag/key` condition.	• 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.	• Assign user or group 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. When IAM Identity Center passes these attributes to the account, they are sent as session tags that you can reference using the `aws:PrincipalTag/`*`tag-key`* condition key in all relevant AWS IAM policy types. 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 between IAM Identity Center and Microsoft AD.

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.

- **Configure attribute mappings in the IAM Identity Center console.** You can map attributes from the IAM Identity Center directory to session tags on the **Attributes for access control** page in the IAM Identity Center console. The attribute values that you choose here are sourced from the Identity Center directory and replace the values for any matching attributes that come from an IdP through a SAML 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. You can then select these synchronized attributes on the **Attributes for access control** page to use them as session tags.

- 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.

- **Pass attributes from your IdP through SAML assertions.** You can configure your IdP to send attributes as session tags through SAML assertions. To do this, configure your IdP to send SAML assertions with the attribute name set to `https://aws.amazon.com/SAML/Attributes/AccessControl:`*`TagKey`*, replacing *`TagKey`* with the session tag key you want to populate. IAM Identity Center passes the attribute name and value from the IdP through for policy evaluation.

  It is not necessary to configure an ABAC attribute mapping on the **Attributes for access control** page for attributes that you pass in through SAML assertions from your external IdP. However, if you configure an ABAC mapping for the same attribute on the **Attributes for access control** page, the mapping from the Identity Center directory takes precedence and replaces the value sent by your IdP in the SAML assertion.

  > **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. For information about how to configure user attributes for access control in your IdP to send through SAML assertions, see the IAM Identity Center identity source tutorials for your IdP.

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 attribute-based access control (ABAC), you must first enable it in either the **Settings** page of the IAM Identity Center console or the IAM Identity Center API. Regardless of the identity source, you can always configure user attributes from the Identity Store for use in ABAC. In the console, you can do this by navigating to the **Attributes for access control** tab on the **Settings** page. If you use an external identity provider (IdP) as the identity source, you also have the option of receiving attributes from the external IdP in SAML assertions. In this case, you need to configure the external IdP to send the desired attributes. If an attribute from a SAML assertion is also defined as an ABAC attribute in IAM Identity Center, IAM Identity Center will send the value from its Identity Store as a session tag on sign-in to an AWS account.

> **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.

Topics

- Enable attributes for access control
- Select your attributes for access control
- Disable attributes for access control

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.

If your IAM Identity Center instance was created before December 2020 and you plan to enable ABAC in it, you must have the `iam:UpdateAssumeRolePolicy` policy associated with the IAM Identity Center administrative role, regardless of whether you have 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.

> **Note**
>
> This procedure applies only when you want to map attributes from your IAM Identity Center directory for use as session tags. If you are passing attributes from an external identity provider (IdP) through SAML assertions, you do not need to configure attribute mappings here. For more information, see Choosing attributes when using an external identity provider as your identity source. Values set in the mapping from the Identity Center directory overwrite values set in SAML assertions.

To select your attributes using the IAM Identity Center console

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 **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
Department	${path:enterprise.department}	✖
CostCenter	${path:enterprise.costCenter}	✖
Add new key	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, lets 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 between IAM Identity Center and External Identity Providers 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 **Manage attributes**.
4. On the **Manage attributes for access control** page, choose **Disable**.
5. In the **Disable attributes for access control** dialog window, review the information and when ready enter **`DISABLE`**, and then choose **Confirm**.

> **Important**
>
> This step deletes all attributes and stops the use of attributes for access control when federating into AWS accounts regardless of whether any attributes are present in SAML assertions from an external identity source provider.

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 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.

JSON

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

When users federate into an AWS account through IAM Identity Center, the configured access control attributes are passed as session tags. You can reference these session tags in your policies using the `aws:PrincipalTag/`*`tag-key`* condition key. This condition key is supported in all relevant AWS IAM policy types where you can use conditions, including identity-based policies, resource-based policies, permissions boundaries, service control policies (SCPs), and VPC endpoint

policies. This enables you to make fine-grained access control decisions based on user attributes across your entire AWS environment.

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.

Understanding service-linked roles in IAM Identity Center

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 Configure 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.

Notes

- If you are 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.

Setting up and using the AWS access portal

The AWS access portal connects your workforce to AWS accounts and cloud applications through IAM Identity Center. Administrators configure the portal and manage user access, while end users sign in once to seamlessly access all their authorized resources.

The AWS access portal provides single sign-on access to:

- AWS accounts in your organization.
- AWS managed applications such as Amazon Quick and Kiro.
- Cloud applications like Office 365, Concur, Salesforce, and others.

When users sign in to the portal, they find the AWS accounts and applications they're authorized to access without additional sign-in.

Getting started with the AWS access portal

For administrators:

You need administrative access to your organization instance or account instance of IAM Identity Center to configure the AWS access portal and manage user access.

1. Optionally customize the AWS access portal URL.
2. Assign user access to AWS accounts and applications. Assigned AWS resources display in the portal.

For end users:

Your administrator must have completed the AWS access portal setup and provided you with your portal URL and sign-in credentials.

1. Get your portal URL from your administrator (typically `https://your-company.awsapps.com/start`).
2. Sign in using the credentials provided by your administrator.
3. Access your resources in your portal.

Configure the AWS access portal

As an administrator, you can customize the AWS access portal to meet your organization's needs and ensure users can easily access their authorized resources.

What you can configure

AWS access portal activation: Set up initial user access to the AWS access portal, including user credential activation and first-time sign-in processes.

Custom AWS access portal URL (optional): Personalize your organization's AWS access portal URL from the default format (`d-xxxxxxxxxx.awsapps.com/start`) to a more recognizable subdomain (`your-company.awsapps.com/start`).

Before you begin

Ensure you have administrative access to IAM Identity Center, verify that IAM Identity Center is set up as either an organization instance or account instance, and plan your custom subdomain name (this is a one-time configuration that cannot be changed later).

Once configured, users can access the AWS access portal using the custom URL and follow the activation process you've established for your organization.

Topics

- Activating the AWS access portal for first-time IAM Identity Center users
- Customizing the AWS access portal URL
- Confirm users can sign in to the AWS access portal

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**, 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 Signing in to the AWS access portal.

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 cannot 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 cannot 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.

Confirm users can sign in to the AWS access portal

The following steps are for IAM Identity Center administrator to confirm that the IAM Identity Center user can sign in to the AWS access portal and access the AWS account.

Sign in to the AWS access portal

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 are 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 are 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.
6. In the **Accounts** tab, locate your AWS account and expand it.
7. The roles available to you are displayed. For example, if you are 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.
8. If you are redirected to the AWS Management Console you successfully finished setting up access to the AWS account.

> **Note**
>
> If you do not see any **AWS accounts** listed, it is 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 or group 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.

Use the AWS access portal

You can launch multiple applications by choosing the AWS account or application tab 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.

How to use the AWS access portal

To use the AWS access portal:

1. **Get your portal URL** from your administrator (typically looks like `https://your-company.awsapps.com/start`).
2. **Sign in** using the credentials provided by your administrator.
3. **Choose the accounts and applications** in your portal that you want to access.

Your administrator controls what you see in the portal based on your role and permissions. Contact your administrator to request additional access in the these situations:

- You do not 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

- 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
- Viewing and ending your active session

Signing in to the AWS access portal

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 following outlines how you can sign in to the AWS access portal, tips for signing in, and how to sign out of the AWS access portal.

Prerequisites

IAM Identity Center needs to be enabled to use the AWS access portal. For more information, see Enable IAM Identity Center.

> **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.

Sign in to the AWS access portal

To sign in to the AWS access portal

1. In your browser window, paste in the sign-in URL that you were provided and choose **Enter**. The URL looks like `d-xxxxxxxxxx.awsapps.com/start` or

your_subdomain.awsapps.com/start. We recommend that you bookmark this link to the portal now so that you can quickly access it later.

2. Sign in using your standard company sign in credentials.

> **Note**
>
> If your administrator sent you an email one-time password (OTP) and this is your first time signing in, enter that password. After you are signed in, you must create a new password for future sign-ins.

If you are prompted for a **Verification code**, check your email and then copy and paste the code into the sign-in page.

> **Note**
>
> Verification codes are typically sent through email, but the delivery method can vary. Check with your administrator for details.

3. Once signed in, you can access any AWS account and application that displays in the portal.

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 will not present an option to enter in an MFA code as long as you are 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 do not 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 cannot 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 are not a robot. Then choose **Next**. You might need to disable ad blocker software if you cannot 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 are 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 Getting started with 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 or group 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 is 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 do not see any **AWS accounts** listed, it is 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 or group 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 is 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 do not see any **AWS accounts** listed, it is 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 or group 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 do not 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 cannot 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 are 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:

IPv4 endpoint:

```
https://[your_subdomain].awsapps.com/start/#/console?
account_id=[account_ID]&role_name=[permission_set_name]&destination=[destination
```

Dual-stack endpoint

```
https://[identity_center_instance_id].portal.[region].app.aws/#/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:

- **IPv4 endpoint:** `https://example.awsapps.com/start/#/console?account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F%2Fconsole.aws.amazon.com%2Fs3%2Fhome`
- **Dual-stack endpoint:** `https://ssoins-1234567890abcdef.portal.us-east-1.app.aws/#/console?account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F%2Fconsole.aws.amazon.com%2Fs3%2Fhome`
- **(AWS GovCloud (US) Region) IPv4 endpoint:** `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`

- **(AWS GovCloud (US) Region) Dual-stack endpoint:** `https://ssoins-1234567890abcdef.portal.us-gov-west-1.app.aws/#/console?account_id=123456789012&role_name=S3FullAccess&destination=https%3A%2F%2Fconsole.amazonaws-us-gov.com%2Fs3%2Fhome`

Registering your device for MFA

For users in the Identity Center directory, use the following procedure within the AWS access portal to register your new device for multi-factor authentication (MFA).

> **Important**
>
> MFA in IAM Identity Center is currently not supported for external identity providers.

Before you begin

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.

Register your device

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 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 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**.

Viewing and ending your active session

You can use your AWS access portal to view the list of your active sessions, and if required, end one or more sessions.

End your active session using your AWS access portal

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 **Security**.
3. On the **Security** page, the number in parentheses next to **Active sessions** indicates how many active sessions you have. Select the check box next to each session that you want to end, and then choose **End sessions**.

> **Tip**
>
> For user background session, you can search for sessions by the Amazon Resource Name (ARN) of the job that is using the session. In the **Session type** list, choose **User background sessions**, and then enter the job ARN in the search box.

You can only end active sessions that are loaded. If you have many sessions, choose **Load more active sessions** to display additional sessions.

4. Select the check box next to each session that you want to end, and then choose **End sessions**.
5. A dialog box appears that confirms you are ending active sessions. Review the information, and if you want to continue, type `confirm`, and then choose **End sessions**.
6. You are returned to your list of active sessions. A green notification message appears to indicate that the selected sessions were successfully ended.

Using IAM Identity Center across multiple AWS Regions

This topic explains how to use AWS IAM Identity Center across multiple AWS Regions. Learn how to replicate your instance to additional Regions, manage workforce access and sessions, deploy applications, and maintain account access during service disruptions.

When you enable an organization instance of IAM Identity Center, you choose a single AWS Region (primary Region). You can replicate this instance to additional AWS Regions if it meets certain prerequisites. IAM Identity Center automatically replicates workforce identities, permission sets, user and group assignments, sessions, and other metadata from the primary Region to the chosen additional Regions.

Benefits of multi-Region support

Replicating IAM Identity Center to additional AWS Regions provides two key benefits:

- **Improved resiliency of AWS account access** – Your workforce can access their AWS accounts even if the IAM Identity Center instance experiences a service disruption in its primary Region. This applies to access with permissions provisioned before the disruption.
- **Enhanced flexibility in choosing deployment Regions for AWS managed applications** – You can deploy AWS managed applications in your preferred Regions to meet application data residency requirements and improve performance through proximity to users. Applications deployed in additional Regions access replicated workforce identities locally for optimal performance and reliability.

Prerequisites and considerations

Before you replicate your IAM Identity Center instance, ensure the following requirements are met:

- **Instance type** - Your IAM Identity Center instance must be an organization instance. Multi-Region support is not available in account instances.
- **Identity source** - Your IAM Identity Center instance must be connected to an external identity provider (IdP), such as Okta. Multi-Region support is not available for instances that use Active Directory or the Identity Center directory as the identity source.
- **AWS Regions** - Multi-Region support is available in commercial Regions enabled by default in your AWS account. Opt-in Regions are not currently supported.

- **KMS key type for encryption at rest** - Your IAM Identity Center instance must be configured with a multi-Region customer managed KMS key. The KMS key must be located in the same AWS account as IAM Identity Center. For more information, see Implementing customer managed KMS keys in AWS IAM Identity Center.
- **AWS managed application compatibility** - Visit the application table in the section called "Applications that you can use with IAM Identity Center" to confirm the following two application requirements:
  - All AWS managed applications that are in use by your organization must support IAM Identity Center that is configured with a customer managed KMS key.
  - The AWS managed applications that you want to deploy in additional Regions must support this type of deployment.
- **External IdP compatibility** - To fully take advantage of multi-Region support, the external IdP must support multiple assertion consumer service (ACS) URLs. This is a SAML feature that is supported by IdPs such as Okta, Microsoft Entra ID, PingFederate, PingOne, and JumpCloud.

  If you use an IdP that doesn't support multiple ACS URLs, such as Google Workspace, we recommend that you work with your IdP vendor to enable this feature. For options that are available without multiple ACS URLs, see the section called "Using AWS managed applications without multiple ACS URLs" and the section called "AWS account access resiliency without multiple ACS URLs".

Choosing an additional Region

When choosing an additional Region among commercial Regions enabled by default, consider these factors:

- **Compliance requirements** – If you need to run AWS managed applications that access datasets limited to a specific Region for compliance reasons, choose the Region where the datasets reside.
- **Performance optimization** – If data residency isn't a factor, select a Region closest to your application users to optimize their experience.
- **Application support** – Verify that your required AWS applications are available in your chosen Region.
- **AWS account access resiliency** – For continuity of access to AWS accounts, choose a Region geographically distant from the primary Region of your IAM Identity Center instance.

> **Note**
>
> IAM Identity Center has a quota on the number of AWS Regions. For more information, see the section called "Additional quotas".

Replicate IAM Identity Center to an additional Region

If your environment meets the prerequisites, such as configuring IAM Identity Center with a multi-Region customer managed KMS key, complete the following steps to replicate your IAM Identity Center instance to an additional Region. Your primary Region continues to operate normally during and after these steps.

Step 1: Create a replica key in the additional Region

Before replicating IAM Identity Center to a Region, you must first create a replica key of your customer managed KMS key in that Region and configure the replica key with the permissions required for the operations of IAM Identity Center. For instructions on creating multi-Region replica keys, see Create multi-Region replica keys.

The recommended approach for the KMS key permissions is to copy the key policy from the primary key, which grants the same permissions already established for IAM Identity Center in the primary Region. Alternatively, you can define Region-specific key policies, though this approach increases the complexity of managing permissions across Regions and may require additional coordination when updating policies in the future.

> **Note**
>
> AWS KMS doesn't synchronize your KMS key policy across the Regions of your multi-Region KMS key. To keep the KMS key policy in sync across the KMS key Regions, you will need to apply changes in each Region individually.

Step 2: Add the Region in IAM Identity Center

Adding a Region in IAM Identity Center triggers automatic replication of IAM Identity Center data to that Region. The replication is asynchronous with eventual consistency. The following tabs provide instructions for doing this in the AWS Management Console and AWS CLI.

Console

To add a Region

1. Open the IAM Identity Center console.
2. In the navigation pane, choose **Settings**.
3. Choose the **Management** tab.
4. In the **Regions for IAM Identity Center** section, choose **Add Region**.
5. In the **AWS Regions available for replication** section, choose your preferred AWS Region. If the Region doesn't appear in the list, it's not available for replication because the KMS key hasn't been replicated there. For more information, see the section called "Implementing customer managed KMS keys".
6. Choose **Add Region**.
7. In the **Regions for IAM Identity Center** section, monitor the Region status. Use the **Refresh** button (circular arrow) to check the latest Region status as needed. After the replication completes, proceed to Step 2.

AWS CLI

To add a Region

```
aws sso-admin add-region \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
    --region-name eu-west-1
```

To check the current Region status

```
aws sso-admin describe-region \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
    --region-name eu-west-1
```

When the Region status is ACTIVE, you can proceed to Step 2.

The duration of the initial replication to an additional Region depends on the amount of data in your IAM Identity Center instance. Subsequent incremental changes are replicated within seconds in most cases.

Step 3: Update external IdP setup

Follow the tutorial for your external IdP in *Identity source tutorials* for the following steps:

Step 3.a: Add the Assertion Consumer Service (ACS) URLs to your external IdP

This step enables direct sign-in to each additional Region and is required to enable sign-in to AWS managed applications deployed in those Regions and for access to AWS accounts through those Regions. To learn where to find the ACS URLs, see ACS endpoints in the primary and additional AWS Regions.

Step 3.b (Optional): Make the AWS access portal available in the external IdP portal

Make the AWS access portal in the additional Region available as a bookmark app in the external IdP portal. Bookmark apps contain only a link (URL) to the desired destination and are similar to a browser bookmark. You can find the AWS access portal URLs in the console by choosing **View all AWS access portal URLs** in the **Regions for IAM Identity Center** section. For more information, see the section called "AWS access portal endpoints in the primary and additional AWS Regions".

IAM Identity Center supports IdP-initiated SAML SSO in each additional Region, but external IdPs typically support this with only a single ACS URL. For continuity, we recommend keeping the primary Region's ACS URL in use for IdP-initiated SAML SSO and relying on bookmark apps and browser bookmarks for access to additional Regions.

Step 4: Confirm firewall and gateway allow-lists

Review your domain allow-lists in firewalls or gateways, and update them based on the documented allow-lists.

Step 5: Provide information to your users

Provide your users with information about the new setup, including the AWS access portal URL in the additional Region and how to use the additional Regions. Review the following sections for relevant details:

- the section called "Workforce access through an additional Region"
- the section called "Failover to an additional Region for AWS account access"
- the section called "Deploying and managing applications across multiple AWS Regions"

Region changes beyond adding the first Region

You can add and remove additional Regions. The primary Region cannot be removed other than by deleting the entire IAM Identity Center instance. For more information on removing a Region, see the section called "Remove a Region from IAM Identity Center".

You cannot promote an additional Region to be the primary or demote the primary Region to be additional.

What data is replicated?

IAM Identity Center replicates the following data:

Data	Replication source and target
Workforce identities (users, groups, group memberships)	From the primary Region to the additional Regions
Permission sets and their assignments to users and groups	From the primary Region to the additional Regions
Configuration (such as external IdP SAML settings)	From the primary Region to the additional Regions
Application metadata and application assignments to users and groups	From an application's connected IAM Identity Center Region to the other enabled Regions
Trusted token issuers	From the primary Region to the additional Regions
Sessions	From the session's originating Region to the other enabled Regions

> **Note**
>
> IAM Identity Center doesn't replicate data stored in AWS managed applications. Also, it doesn't change the regional footprint of an application deployment. For example, if your

IAM Identity Center instance in in US East (N. Virginia), and you have Amazon Redshift deployed in the same Region, replicating IAM Identity Center to US West (Oregon) doesn't affect the deployment Region of Amazon Redshift and the data it stores.

Considerations:

- **Global resource identifiers across enabled Regions** - Users, groups, permission sets, and other resources have the same identifiers across the enabled Regions.
- **Replication doesn't affect provisioned IAM roles** - Existing IAM roles provisioned from permission set assignments are used during account sign-in from any enabled Region.

Workforce access through an additional Region

This section explains how your workforce can access the AWS access portal, AWS accounts, and applications when you have enabled IAM Identity Center in multiple Regions.

The AWS access portal in an additional Region displays the AWS accounts and applications your workforce has access to in the same way as in the primary Region. Your workforce can sign into the AWS access portal in an additional Region through a direct link to the regional portal endpoint (for example, `https://ssoins-111111h2222j33pp.eu-west-1.portal.amazonaws.com`) or through a bookmark app you set up in the external IdP.

You can use the AWS access portal endpoint in an additional Region to authorize the AWS CLI for access to APIs as an IAM Identity Center user. This functionality works in the same way as in the primary Region. However, CLI authorizations are not replicated across enabled Regions. Therefore, you have to authorize the CLI in each Region individually.

User sessions across multiple AWS Regions

IAM Identity Center replicates user sessions from the originating Region to the other enabled Regions. Session revocation and sign-out in one Region are also replicated to the other Regions.

Session revocation by IAM Identity Center administrators

IAM Identity Center administrators can revoke user sessions in additional Regions. As sessions are replicated across Regions, under normal conditions it suffices to revoke a session in a single

Region and let IAM Identity Center replicate the change to the other enabled Regions. If the primary Region of IAM Identity Center has a disruption, administrators can perform this operation in additional Regions.

AWS access portal endpoints in the primary and additional AWS Regions

If you need to look up the AWS access portal URLs for the enabled Regions, follow these steps:

1. Open the IAM Identity Center console.
2. In the navigation pane, choose **Settings**.
3. Choose the **Management** tab.
4. In the **Regions for IAM Identity Center** section, choose **View all AWS access portal URLs**.

The following table specifies the AWS access portal endpoints across the primary and additional Regions of an IAM Identity Center instance.

AWS access portal endpoint	Primary Region	Additional Region	URL pattern and example
Classic IPv4 only[1]	Yes	No	**Pattern:** `https://[Identity Store ID].awsapps.com/start` **Example:** `https://d-12345678.awsapps.com/start`
Custom-alias IPv4 only[1]	Yes (optional)	No	**Pattern:** `https://[custom alias].awsapps.com/start` **Example:** `https://mycompany.awsapps.com/start`
Alternative IPv4 only[2]	Yes	Yes	**Pattern:** `https://[Identity Center instance ID].`

AWS access portal endpoint	Primary Region	Additional Region	URL pattern and example
			[Region].portal.amazonaws.com **Example:** https://ssoins-111111h2222j33pp.eu-west-1.portal.amazonaws.com
Dual-stack[2]	Yes	Yes	**Pattern:** https://*[Identity Center instance ID]*.portal.*[Region]*.app.aws **Example:** https://ssoins-111111h2222j33pp.portal.eu-west-1.app.aws

[1] In additional Regions, the custom alias is not supported, and the `awsapps.com` parent domain is not available.

[2] The alternative IPv4 only and dual-stack portal endpoints don't have the trailing `/start` in the URL.

Assertion Consumer Service (ACS) endpoints in the primary and additional AWS Regions

If you need to look up the ACS URLs or download them as part of the SAML metadata, follow these steps:

1. Open the IAM Identity Center console.
2. In the navigation pane, choose **Settings**.
3. Choose the **Identity source** tab.
4. In the **Actions** dropdown menu, choose **Manage authentication**.

5. The **Service provider metadata** section displays the AWS access portal and ACS URL for each enabled Region. IPv4-only and dual-stack URLs are displayed on separate tabs. If your IdP supports uploading a SAML metadata file, you can choose **Download metadata file** to download the SAML metadata file with all ACS URLs. If your IdP does not support uploading a metadata file, or you prefer to add ACS URLs individually, you can copy individual URLs from the table in the console, or choose **View ACS URLs** and then **Copy all URLs**. Retain the ACS URL that is already configured for the primary Region to ensure that service provider-initiated SAML single sign-on from the primary Region remains functional.

The following table specifies the SAML Assertion Consumer Service (ACS) endpoints across the primary and additional Regions of an IAM Identity Center instance:

ACS endpoint	Primary Region	Additional Region	URL pattern and example
IPv4 only	Yes	Yes	**Pattern:** `https://[Region].signin.aws/platform/saml/acs/[Tenant ID]` **Example:** `https://us-west-2.signin.aws/platform/saml/acs/1111111111111111-aaee-ffff-dddd-11111111111`
Alternative IPv4 only*	Yes	No	**Pattern:** `https://[Region].signin.aws.amazon.com/platform/saml/acs/[Tenant ID]` **Example:** `https://us-west-2.signin.aws.amazon.com/platform/saml/acs/1111111111111111-aaee-ffff-dddd-11111111111`
Dual-stack	Yes	Yes	**Pattern:** `https://[Region].sso.sign`

ACS endpoint	Primary Region	Additional Region	URL pattern and example
			`in.aws/platform/saml/acs/`*`[Tenant ID]`* **Example:** `https://us-west-2.sso.signin.aws/platform/saml/acs/1111111111111111-aaee-ffff-dddd-11111111111`

* For instances that were enabled before February 2026, retain this endpoint because service provider-initiated SAML single sign-on to the primary Region requires it. IAM Identity Center does not use this endpoint for instances that were enabled in February 2026 or later.

Using AWS managed applications without multiple ACS URLs

Some external identity providers (IdPs) don't support multiple assertion consumer service (ACS) URLs in their IAM Identity Center application. Multiple ACS URLs are a SAML feature that is required for direct sign-in to a specific Region in a multi-Region IAM Identity Center.

For example, if you launch an AWS managed application through an application link, the system triggers sign-in through the application's connected IAM Identity Center Region. However, if the ACS URL for that Region is not configured in the external IdP, the sign-in fails.

To resolve this issue, work with your IdP vendor to enable support for multiple ACS URLs. In the meantime, you can still use AWS managed applications in additional Regions. First, sign into the Region whose ACS URL is configured in the external IdP (the primary Region by default). After you have an active session in IAM Identity Center, you can launch the application from the AWS access portal in any enabled Region, or through an application link.

Failover to an additional Region for AWS account access

The topic of AWS account access through IAM Identity Center is covered extensively in *AWS account access*. This section provides additional details relevant to maintaining AWS account access across multiple AWS Regions in the event of a service disruption in the primary Region.

If your IAM Identity Center instance experiences a disruption in the primary Region (for example, the AWS access portal in the primary Region is unavailable), your workforce can switch to an

additional Region to continue accessing AWS accounts and unaffected applications. For more information, see the section called "Workforce access through an additional Region".

We recommend that you communicate the AWS access portal endpoints in additional Regions and the external IdP setup (such as bookmark apps for the additional Regions) to your workforce as soon as you complete the setup in the section called "Replicate IAM Identity Center to an additional Region". This will enable them to be ready for failover to an additional Region if needed.

Similarly, we recommend that AWS CLI users create AWS CLI profiles for additional Regions for each of the profiles they have for the primary Region. Then, they can switch to that profile if there is a service disruption in the primary Region.

> **Note**
>
> Continuity of access to AWS accounts also depends on the health of your external IdP and permissions such as permission set assignments and group memberships being provisioned and replicated before a service disruption. We recommend your organization also set up AWS break-glass access to maintain AWS access to a small group of privileged users when the external IdP has a service disruption. Emergency access is a similar option that avoids using IAM users, but it too depends on the external IdP.

AWS account access resiliency without multiple ACS URLs

Some external identity providers (IdPs) don't support multiple assertion consumer service (ACS) URLs in their IAM Identity Center application. Multiple ACS URLs are a SAML feature that is required for direct sign-in to a specific Region in a multi-Region IAM Identity Center.

To enable your users to access their AWS accounts through multiple IAM Identity Center Regions, you must configure the respective regional ACS URLs in the external IdP. However, if the external IdP supports only a single ACS URL in their IAM Identity Center application, users can directly sign into a single IAM Identity Center Region.

To resolve this issue, work with your IdP vendor to enable support for multiple ACS URLs. In the meantime, you can use additional Regions as backup for access to AWS accounts.

If an IAM Identity Center service disruption occurs in the primary Region, you must update the ACS URL in the external IdP with an additional Region's ACS URL. After this update, your users

can access the AWS access portal in the additional Region using the existing IAM Identity Center application in the external IdP portal, or through a direct link that you share with them.

We recommend that you test this setup periodically to ensure that it works when needed and communicate this failover process to your organization.

> **Note**
>
> When you use an additional Region for access to AWS accounts in this setup, your users might not be able to access AWS managed applications that are connected to the primary Region. Therefore, we recommend this only as a temporary measure to maintain access to AWS accounts.

Deploying and managing applications across multiple AWS Regions

The topic of application access through IAM Identity Center is covered extensively in *Application access*. This section provides additional details relevant to the deployment and management of applications across multiple AWS Regions.

Deploying and managing AWS managed applications across multiple AWS Regions

With a single-Region IAM Identity Center instance, you can deploy AWS managed applications in the same Region as your instance. Some applications such as Amazon Q Business support a cross-Region connection to IAM Identity Center, which enables their deployment outside the IAM Identity Center's Region if the application of interest is available there. However, cross-Region calls can cause slower application performance, and most AWS managed applications don't support this type of connection.

A multi-Region IAM Identity Center instance lets you deploy AWS managed applications in any enabled Region with a connection to IAM Identity Center in the same Region ("Region-local connection"). Deployment in the primary Region is supported by all integrated AWS managed applications. To confirm which AWS managed applications support deployment in additional Regions of IAM Identity Center, see the applications table in the section called “Applications that you can use with IAM Identity Center”. In any case, the AWS managed application has to be available in the Region where you want to deploy it.

With a Region-local connection to IAM Identity Center, AWS managed applications access workforce identities in the same Region for optimal performance and reliability. We recommend choosing a Region-local connection when deploying an AWS managed application whenever the prerequisites are met.

To deploy an AWS managed application in an additional Region of IAM Identity Center, start the deployment in that Region through the application console or API in the same way that you deploy in the primary Region.

Considerations:

- If you haven't replicated your IAM Identity Center to that Region yet, we recommend that you do this first so that the application deployment can complete right away.
- AWS managed applications will, in many cases, automatically establish a Region-local connection if you've already replicated IAM Identity Center to the Region.
- If an AWS managed application offers a cross-Region connection to IAM Identity Center, we recommend that you choose a Region-local connection provided that the prerequisites are met.
- If the application doesn't support deployment in additional Regions, you can deploy it in the primary Region provided that the application is available there.

> **Important**
>
> If your IAM Identity Center instance is multi-regional, all AWS managed applications in use by your organization must support IAM Identity Center configured with a customer-managed KMS key regardless of the application deployment Region. Confirm this in the the section called “Applications that you can use with IAM Identity Center” before deploying an application and before configuring a customer-managed KMS key in your IAM Identity Center.

An application's management Region

After you deploy an AWS managed application in an additional Region of IAM Identity Center using a Region-local connection, you manage the application and its assignments to users and groups in the same Region. IAM Identity Center replicates the application metadata including assignments to users and groups to other enabled Regions so that your workforce can launch applications from any enabled Region.

If your AWS managed application is using a cross-Region connection to IAM Identity Center, you can manage the application details such as name and description, and application assignments to users and groups through IAM Identity Center console and API in the connected Region. Regardless of the connection type, you can manage the application through its console in its deployment Region.

Trusted identity propagation

You can use trusted identity propagation with AWS managed applications that support it in any enabled Region of your IAM Identity Center instance.

All applications that propagate identity context to each other must be in the same Region.

An application's dependency on its connected IAM Identity Center Region

Each AWS managed application connects to a specific IAM Identity Center Region during deployment. The application then depends on that Region for user sign-in, even if your IAM Identity Center is enabled in multiple Regions. If your IAM Identity Center is experiencing a disruption in that Region, users might not be able to access AWS managed applications connected to the Region.

Deploying and managing customer managed applications across multiple AWS Regions

IAM Identity Center supports SAML and OAuth2 customer managed applications. You can choose to create them in any enabled Region of your IAM Identity Center instance. After you create one, you manage the application and its assignments to users and groups in the same Region.

Remove a Region from IAM Identity Center

To remove an additional Region from your IAM Identity Center instance, follow these steps:

Step 1: Update external IdP configuration

You can choose to remove the ACS URL for this Region from your external IdP or keep it in case you want to add this Region again later. We recommend that you remove or hide the bookmark app you might have created for the AWS access portal in this Region.

Step 2: Remove the Region

Console

To add a Region

1. Open the IAM Identity Center console.
2. In the navigation pane, choose **Settings**.
3. Choose the **Management** tab.
4. In the **Regions for IAM Identity Center** section, choose the additional Region you want to remove.
5. Choose **Remove**.
6. Before confirming removal by choosing **Remove Region**, pay attention to the warning about the potential loss of access to applications that were created in this IAM Identity Center Region. If you're not sure whether you have such applications, choose **Applications** in the navigation pane and confirm the connected Region in the **Created from** column for each AWS managed and customer managed application.

   > **Note**
   >
   > You may continue incurring charges for deployments of AWS managed applications that are still connected to the removed Region even if you lose access to these applications. To prevent this, you need to remove these AWS managed application deployments through the application console or API before removing the Region in IAM Identity Center. If you already removed the IAM Identity Center Region, you can restore access to applications by adding the Region back.

7. In the **Regions for IAM Identity Center** section, monitor the Region status. Use the **Refresh** button (circular arrow) to see the latest Region status as needed. After the Region is removed, the Region no longer appears in the Region list.

AWS CLI

To remove a Region

```
aws sso-admin remove-region \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
```

```
    --region-name eu-west-1
```

To check the current Region status

```
aws sso-admin describe-region \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
    --region-name eu-west-1
```

When the Region is removed, proceed to Step 2.

Step 3: Delete the replica key

You can choose to remove the replica key from this Region to avoid incurring KMS storage charges. For more information, see Deleting an AWS KMS key.

> **Important**
>
> Make sure to delete only the replica key in this specific Region. The other IAM Identity Center Regions continue to rely on the KMS key in the other enabled Regions for normal operations.

IAM Identity Center service APIs supported in additional AWS Regions

API	Functionality in additional Regions
Identity Center API	**Supported:** • Application management read/write operations • Instance read operations **Not supported:** • Any write operations other than application related • Permission set and account assignment read operations

API	Functionality in additional Regions
Identity Store API	Read operations
OIDC API	All operations are supported
Access Portal API	All operations are supported
SCIM API	No operation is supported

AWS CloudTrail events of IAM Identity Center in the primary and additional Region

In AWS CloudTrail, you can audit actions that IAM Identity Center users and administrators take across all enabled AWS Regions of your IAM Identity Center instance. The AWS CloudTrail events are emitted and logged in the Region where the action takes place. For more information, see the section called "Logging and monitoring".

Availability of IAM Identity Center use cases in the primary and additional Regions

Feature	Regional availability
Workforce directory with a user portal	
User access to the AWS access portal including portal sign-in and global sessions (one sign-in for all Regions)	All enabled Regions
Display of all assigned accounts	All enabled Regions
Display of all assigned applications (regardless of where the applications were created)	All enabled Regions
Read access to users, groups, and memberships in the AWS Console or via Identity Store APIs	All enabled Regions

Feature	Regional availability
Revoke user sessions	All enabled Regions
Automatic synchronization of users and groups from an external identity source such as external IdP through SCIM API or Identity Store API	Primary Region only
Configure automatic identity provisioning with SCIM	Primary Region only
Configure SAML SSO with an external IdP	Primary Region only - read access through the console in all enabled Regions
Create/update/delete operations on users, groups and group memberships via the console or Identity Store APIs.	Primary Region: available via Identity Store API but blocked in the IAM Identity Center console when SCIM API is used for provisioning (except disable/enable user access and delete user, which are always available). Additional Regions: unavailable
Multi-account access	
Access assigned accounts via the AWS access portal, AWS CLI, and shortcut links	All enabled Regions
Manage multi-account permission sets and their assignments in the console and APIs (including temporary elevated access)	Primary Region only
Access to applications and AWS services	
Deploy AWS managed applications through the application console and APIs	All enabled Regions – subject to applications' regional availability and support for deployment in additional Regions

Feature	Regional availability
Create customer managed applications through the Identity Center console and APIs	All enabled Regions
Manage application metadata and assignments in the console and APIs	Application's connected IAM Identity Center Region
Launch applications from the AWS access portal or directly via an application link or bookmark	All enabled Regions
SSO to Amazon EC2 instances	All enabled Regions
Trusted identity propagation	
Create a trusted token issuer	Primary Region only
Trusted identity propagation with AWS managed applications	All enabled Regions - Applications that propagate identity context to each other must be in the same Region
Other administrative features	
All other administrative features such as Region management, KMS key management, instance management, and session management (except session revocation)	Primary Region only - read access available in all enabled Regions for some data (permission set assignments excluded)

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 can replicate your IAM Identity Center instance to additional Regions to maintain account access with already provisioned permissions in the event of a Regional disruption. For more information, see *Using IAM Identity Center across multiple AWS Regions*.

You enable IAM Identity Center in your AWS Organizations management account. This is required for IAM Identity Center to provision, deprovision, and update roles across all your AWS accounts. When you enable IAM Identity Center, it is deployed to the AWS Region that is currently selected, referred to as the "primary Region". If you want to deploy to a specific AWS Region, change the Region selection before enabling IAM Identity Center because the primary Region cannot be changed after IAM Identity Center is enabled.

IAM Identity Center supports most administrative functions only from the primary Region. This includes the connection to an external identity provider, synchronization of users and groups, and the creation and assignment of permission sets to users and groups. In contrast, the management of applications and their assignments must take place in the IAM Identity Center Region where the application was created.

> **Note**
>
> Even if your IAM Identity Center is replicated to additional Regions, we recommend that you set up AWS break-glass access. This helps you maintain AWS access for a small group of privileged users during events such as a service disruption in the external IdP. Emergency access is another option that uses identities from an external IdP instead of IAM users; however, it does not protect against a disruption in the external IdP.

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 AI, however, must be installed in a Region of your IAM Identity Center instance 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 and the section called "Deploying and managing AWS managed applications across multiple AWS Regions".

You can also use IAM Identity Center to authenticate and authorize access to SAML-based customer managed applications that are reachable through a public URL, regardless of the platform or cloud on which the application is built.

We do not recommend using Account instances of IAM Identity Center as a means to implement resiliency because they do not support AWS account access and because they create 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 in a single AWS Region. 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.

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 . If you prefer to not depend on the external IdP, consider setting up AWS break-glass access

> **Important**
>
> We recommend that you deploy this configuration before a disruption occurs because you cannot 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.

Topics

- 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
- One-time setup of a direct IAM federation application with ADFS

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. This account will become your emergency operations account.
2. Connect your IdP to the emergency operations account by using SAML 2.0-based federation.

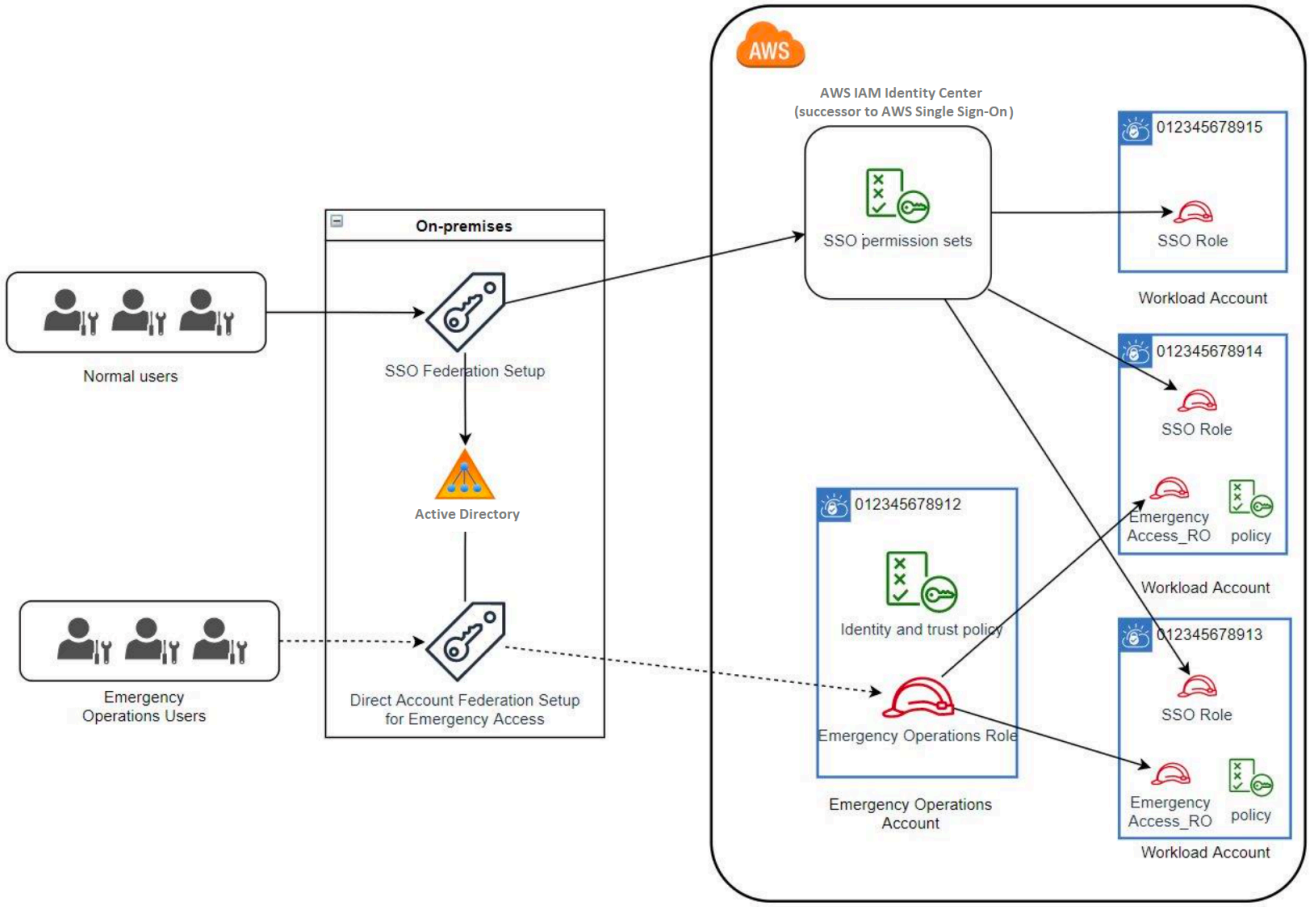
3. In the emergency operations account, create a role for third-party identity provider federation. Also, create an emergency operations role in each of your workload accounts, with your required permissions.
4. Delegate access to your workload accounts for the IAM role that you created in the emergency 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

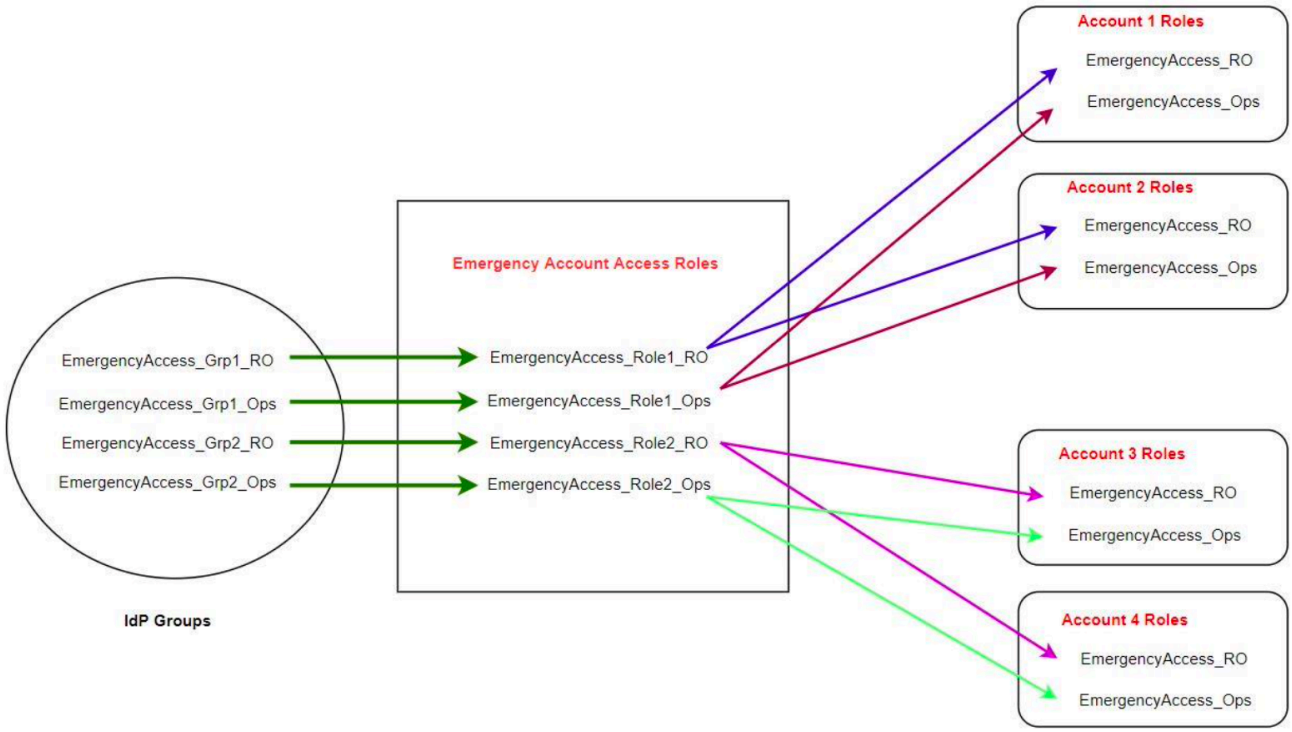
Before you configure emergency access, create a plan for how the access model will work. Use the following process to create this plan.

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. If you are using Okta or other external IdPs as your identity source, see One-time setup of a direct IAM federation application in Okta. If you are using Active Directory as your identity source, see One-time setup of a direct IAM federation application with ADFS.
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. To handle regional failure scenarios, when you configure the sign-in endpoint, we recommend that you enable both the non-regional endpoint and multiple regional endpoints for all Regions you operate in to improve federation resiliency. When configuring the ACS URL for this emergency access, we recommend that you use the regional endpoint of a different Region than the one that your IAM Identity Center is deployed in. Refer to Sign-in endpoints in the *AWS General Reference* for the list of regional endpoints.
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	^aws\#\S+\#(?{{role}}[\w\-]+)\#(?{{accountid}}\d+)$
Role Value Pattern	arn:aws:iam::${accountid}:saml-provider/Okta,arn:aws:iam::${accountid}:role/${role}

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_RO` 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_RO#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_RO#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_RO` role. For an illustration, see the emergency account in the diagram under How to design emergency role, account, and group mapping . Replace the sample SAML provider ARN with the correct one you have created in the emergency access account. Replace the regional endpoints in the example with those regions of your choice.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
         "Effect":"Allow",
         "Principal":{
            "Federated":"arn:aws:iam::123456789012:saml-provider/[SAML
 PROVIDER NAME]"
         },
         "Action":[
            "sts:AssumeRoleWithSAML",
            "sts:TagSession"
         ],
```

```
         "Condition":{
            "StringEquals":{
               "SAML:aud": [
                        "https://signin.aws.amazon.com/saml",
                        "https://us-west-2.signin.aws.amazon.com/saml",
                        "https://us-west-1.signin.aws.amazon.com/saml",
                        "https://us-east-2.signin.aws.amazon.com/saml"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Principal":{
         "Federated":"arn:aws:iam::123456789012:saml-provider/Okta"
         },
         "Action":"sts:SetSourceIdentity"
       }
   ]
}
```

8. The following is an example statement for a **permissions policy** that is attached to the `EmergencyAccess_Role1_RO` role. For an illustration, see the emergency account in the diagram under How to design emergency role, account, and group mapping .

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/EmergencyAccess_RO",
                "arn:aws:iam::444455556666:role/EmergencyAccess_RO"
            ]
        }
    ]
}
```

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 .

   JSON

```
{
    "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.

One-time setup of a direct IAM federation application with ADFS

This guide describes the one-time setup process for configuring direct IAM federation with ADFS to enable emergency access to AWS accounts when IAM Identity Center is unavailable.

Prerequisites

If you plan to configure ADFS with AWS Managed Microsoft AD, we recommend that you first configure Multi-Region replication and continue with the following steps in the additional Region and not in the primary Region for resiliency.

Plan your Active Directory group naming convention

Create AD groups using a specific naming pattern that enables automated matching between group names and AWS IAM roles.

Group naming format: `AWS-<AccountNumber>-<RoleName>`

For an illustration, see the emergency account in the diagram under How to design emergency role, account, and group mapping . When a user is assigned to this group, they are granted access to the `EmergencyAccess_Role1_R0` role in account `123456789012`. If a user is associated with multiple groups, they see a list of available roles by AWS account and can choose which role to assume.

AWS configuration

The complete setup includes configurations in an emergency access account and the workload accounts. For an illustration of the overall setup, see How to design emergency role, account, and group mapping .

1. Create a SAML identity provider
2. Create emergency access roles
3. Configure workload account roles

Create a SAML identity provider

In the emergency access account, create a SAML identity provider in IAM by following the steps in Create a SAML identity provider in IAM. Download the required metadata from your ADFS server:

```
https://<yourADFSserverFQDN>/FederationMetadata/2007-06/
FederationMetadata.xml
```

Create emergency access roles

Create emergency access roles in the emergency account using SAML 2.0 Federation as the trusted entity type. Select the SAML 2.0 provider you created in the previous step.

Considerations:

- **Include all Regions where you operate** — select every Region in which you have active workloads to ensure federation remains available during a Regional disruption.

- **Configure at least one additional Regional endpoint, even if you operate in a single Region** — for example, if you operate only in `us-east-1`, add `us-west-2` as a secondary endpoint. You can fail over your IdP to the `us-west-2` SAML sign-in endpoint and still access your `us-east-1` resources, even without any workloads in `us-west-2`.
- **Enable both the non-regional endpoint and regional endpoints** — Although the non-regional endpoint (`https://signin.aws.amazon.com/saml`) is highly available, it is hosted in a single AWS Region, `us-east-1`, while regional endpoints (`https://<region>.signin.aws.amazon.com/saml`) improve resiliency by reducing dependency on a single global endpoint.

Configure trust policy

Refer to One-time setup of a direct IAM federation application in Okta for an example trust policy with multiple sign-in regional endpoints. Replace the sample regional endpoints and SAML provider ARNs with yours.

Configure permissions policies

Refer to One-time setup of a direct IAM federation application in Okta for an example permissions policy that you attach to the emergency access roles.

Configure workload account roles

For the workload account roles, configure a custom trust policy that allows the emergency access roles in the emergency access account to assume them. Refer to One-time setup of a direct IAM federation application in Okta for an example trust policy, where account `123456789012` is the emergency access account.

Active Directory configuration

The following steps describe how to configure Active Directory and ADFS for emergency access.

1. Create groups
2. Create a relying party
3. Create claim rules

Create groups

Create emergency groups in Active Directory according to the naming convention described earlier (for example, `AWS-123456789012-EmergencyAccess_Role1_RO`). Assign users to these groups through your existing provisioning mechanisms.

Create a relying party

ADFS federation requires a relying party configuration. The relying party is AWS Security Token Service (AWS STS), which outsources authentication to ADFS as the identity provider.

1. In the ADFS Management console, use the Action menu and select **Add Relying Party Trust**. Select **Claims aware** when adding a relying party.
2. For federation metadata, enter the Metadata URL from the identity provider metadata info on the IAM console. For example:

   ```
   https://signin.aws.amazon.com/static/saml/SAMLSPXXXXXX/saml-metadata.xml
   ```
3. Set the display name for the relying party (for example, **AWS Account Access**) and then choose **Next**.
4. Select who you want to enable to access AWS. You can select specific groups and define requirements like MFA.
5. Choose **Close** on the Finish page to complete the Add Relying Party Trust Wizard. AWS is now configured as a relying party.

Create claim rules

ADFS uses Claims Rule Language to issue and transform claims between claims providers and relying parties. You need to create four claim rules: NameId, RoleSessionName, Get AD Groups, and Roles for AWS Access.

Right-click on the relying party and then choose **Edit Claim Issuance Policy**. Choose **Add Rule** to add rules.

1. NameId

a. Select **Transform an Incoming Claim** and then choose **Next**.

b. Use the following settings:

- Claim rule name: `NameId`
- Incoming claim type: `Windows Account Name`
- Outgoing claim type: `Name ID`
- Outgoing name ID format: `Persistent Identifier`
- Pass through all claim values: checked

c. Choose **OK**.

2. RoleSessionName

a. Choose **Add Rule**.

b. In the Claim rule template list, select **Send LDAP Attributes as Claims**.

c. Use the following settings:

- Claim rule name: `RoleSessionName`
- Attribute store: `Active Directory`
- LDAP Attribute: `E-Mail-Addresses`
- Outgoing Claim type: `https://aws.amazon.com/SAML/Attributes/RoleSessionName`

d. Choose **OK**.

3. Get AD Groups

a. Choose **Add Rule**.

b. In the Claim rule template list, select **Send Claims Using a Custom Rule** and then choose **Next**.

c. For Claim Rule Name, enter `Get AD Groups`, and then in Custom rule, enter the following:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/
windowsaccountname", Issuer == "AD AUTHORITY"]
=> add(store = "Active Directory", types = ("http://temp/variable"), query =
 ";tokenGroups;{0}", param = c.Value);
```

This custom rule uses a script in the claim rule language that retrieves all the groups the authenticated user is a member of and places them into a temporary claim named `http://temp/variable`.

> **Note**
>
> Ensure there's no trailing whitespace to avoid unexpected results.

4. Role Attributes

a. Choose **Add Rule**.

b. In the Claim rule template list, select **Send Claims Using a Custom Rule** and then choose **Next**.

c. For Claim Rule Name, enter `Roles`, and then in Custom rule, enter the following:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-([\d]{12})"]
=> issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value =
 RegExReplace(c.Value, "AWS-([\d]{12})-", "arn:aws:iam::$1:saml-provider/
<ADFS>,arn:aws:iam::$1:role/"));
```

This custom rule uses regular expressions to transform each of the group memberships of the form `AWS-<Account Number>-<Role Name>` into the IAM role ARN and IAM federation provider ARN form that AWS expects.

> **Note**
>
> In the example rule language above, `ADFS` represents the logical name given to the SAML identity provider in the AWS identity provider setup. Change this based on the logical name you chose in the IAM console for your identity provider.

Test the configuration

Test that the solution works by authenticating at `https://<yourADFSserverFQDN>/adfs/ls/IdpInitiatedSignOn.aspx`. Select the name of the relying party you created from the dropdown list of the sites.

Update default SAML assertion endpoint in ADFS

> **Important**
>
> When configuring relying party trust in ADFS, the SAML Assertion endpoint defaults to `https://signin.aws.amazon.com/` which is not a global endpoint and is located in `us-east-1`. We recommend that you modify the default endpoint to a regional endpoint different from where your IAM Identity Center is configured for resiliency. For example, if your IAM Identity Center is deployed in `us-east-1` and you also operate in `us-west-2`, change the default SAML Assertion consumer endpoint to `https://us-west-2.signin.aws.amazon.com/saml`.

1. Choose **Properties** on the relying party trust and go to the **Monitoring** tab. Clear the checkbox **Automatically Update relying party**.
2. Go to the **Endpoints** tab, select your preferred sign-in endpoint, and choose **Edit**.
3. Select the checkbox **Set the trusted URL as default**. Choose **OK** and **Apply** for the setting to take effect.

> **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 cannot create or access IAM Identity Center resources. For example, you must have permissions to create an IAM Identity Center connected directory.

> **Note**
>
> If your IAM Identity Center instance is configured with a customer managed KMS key, your IAM Identity Center administrators and other actors who need access to the KMS key will

need additional permissions. Refer to Implementing customer managed KMS keys in AWS IAM Identity Center.

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
- Resource-based policy example for IAM Identity Center 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.

JSON

```
{
   "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 do not 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 do not 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.

> **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.

JSON

```
{
    "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:ListDelegatedAdministrators",
                "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.

JSON

```
{
    "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.

JSON

```
{
    "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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sso-directory:ListGroupsForUser",
                "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 will not 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sso:DescribeAccountAssignmentCreationStatus",
                "sso:DescribeAccountAssignmentDeletionStatus",
                "sso:DescribePermissionSet",
                "sso:DescribePermissionSetProvisioningStatus",
                "sso:DescribeRegisteredRegions",
                "sso:GetApplicationInstance",
                "sso:GetApplicationTemplate",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:GetManagedApplicationInstance",
                "sso:GetMfaDeviceManagementForDirectory",
                "sso:GetPermissionSet",
                "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": "*"
        }
    ]
}
```

Resource-based policy example for IAM Identity Center IAM Identity Center

Every application that works with IAM Identity Center and uses OAuth 2.0 requires a resource-based policy. The application can be customer managed or AWS managed. The required resource-based policy, called the *application policy* (or ActorPolicy in the APIs), defines which IAM principals are authorized to call IAM authentication method API actions such as `CreateTokenWithIAM`. The IAM authentication method allows an IAM principal, such an IAM role or an AWS service, to authenticate to the IAM Identity Center OIDC service by presenting IAM credentials to request or manage access tokens at the **/token?aws_iam=t** endpoint.

The application policy governs operations for issuing tokens (`CreateTokenWithIAM`). The policy also governs permission-only actions that are used only by AWS managed applications for validating tokens (`IntrospectTokenWithIAM`) and revoking tokens (`RevokeTokenWithIAM`). For a customer managed application, you configure this policy by specifying which IAM principals are authorized to call `CreateTokenWithIAM`. When an authorized principal calls this API action, the principal receives access and refresh tokens for the application.

If you are using the IAM Identity Center console to set up a customer managed application for trusted identity propagation, see Step 4 in Set up customer managed OAuth 2.0 applications for

information about how to configure the application policy. For an example policy, see Example policy: Allow an IAM role to create access and refresh tokens later in this topic.

Policy requirements

The policy must meet the following requirements:

- The policy must include a `Version` element set to "2012-10-17".
- The policy must include at least one `Statement` element.
- Each policy `Statement` must include the following elements: `Effect`, `Principal`, `Action`, and `Resource`.

Policy elements

The policy must include the following elements:

Version

Specifies the policy document version. Set the version to `2012-10-17` (the latest version).

Statement

Contains the policy `Statements`. The policy must contain at least one `Statement`.

Each policy `Statement` consists of the following elements.

Effect

(Required) Determines whether to allow or deny the permissions in the policy statement. Valid values are `Allow` or `Deny`.

Principal

(Required) The principal is the identity that gets the permissions specified in the policy statement. You can specify IAM roles or AWS service principals.

Action

(Required) The IAM Identity Center OIDC service API operations to allow or deny. Valid actions include:

- `sso-oauth:CreateTokenWithIAM`: This action, which corresponds to the `CreateTokenWithIAM` API operation, grants permission to create and return access and

refresh tokens for authorized client applications that are authenticated using any IAM entity, such as an AWS service role or user. These tokens might contain defined scopes that specify permissions such as `read:profile` or `write:data`.

- `sso-oauth:IntrospectTokenWithIAM` [permission only]: Grants permission to validate and retrieve information about active OAuth 2.0 access tokens and refresh tokens, including their associated scopes and permissions. This permission is used only by AWS managed applications and is not documented in the *IAM Identity Center OIDC API Reference*.
- `RevokeTokenWithIAM` [permission only]: Grants permission to revoke OAuth 2.0 access tokens and refresh tokens, invalidating them before their normal expiration. This permission is used only by AWS managed applications and is not documented in the *IAM Identity Center OIDC API Reference*.

Resource

(Required) In this policy, the value of the `Resource` element is `"*"`, which means "this application."

For more information about AWS policy syntax, see AWS IAM Policy Reference in the *IAM User Guide*.

Example policy: Allow an IAM role to create access and refresh tokens

The following permissions policy grants permissions to `ExampleAppClientRole`, an IAM role assumed by a workload, to create and return access and refresh tokens.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRoleToCreateTokens",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleAppClientRole"
            },
            "Action": "sso-oauth:CreateTokenWithIAM",
            "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.

This policy also includes AWS Key Management Service permissions required for IAM Identity Center instances that use customer managed keys for encryption.

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 Configure 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.
- `AllowKMSKeyUseViaService` and `AllowKMSKeyDiscovery` – Allows AWS Key Management Service operations for customer managed keys used by IAM Identity Center instances.

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`.

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.

This policy also includes AWS Key Management Service permissions required for IAM Identity Center instances that use customer managed keys for encryption.

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 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 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 polices to them. The IAM Identity Center administrator must have the permissions to list the existing policies so that they can choose which polices 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. This policy also includes AWS Key Management Service permissions required for IAM Identity Center instances that use customer managed keys for encryption.

This policy includes the following permissions:

- **IAM Identity Center Directory** - Full administrative access to IAM Identity Center directory operations.
- **Identity Store** - Full administrative access to identity store operations and authentication.
- **IAM Identity Center** - Permission to list directory associations.
- **AWS Key Management Service** - Permissions to decrypt, describe keys, and generate data keys for customer managed keys used by IAM Identity Center instances.

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.

This policy also includes AWS Key Management Service permissions required for IAM Identity Center instances that use customer managed keys for encryption.

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 cannot make any updates in IAM Identity Center. For example, principals with these permissions can view IAM Identity Center users, but they cannot change any user attributes or assign MFA devices.

This policy also includes AWS Key Management Service permissions required for IAM Identity Center instances that use customer managed keys for encryption.

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 cannot

make any updates to synchronization settings. For example, principals with these permissions can view identity synchronization settings, but cannot 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 cannot 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*.

AWS managed policy: AWSIdentityCenterExternalManagementPolicy

You can attach the `AWSIdentityCenterExternalManagementPolicy` policy to your IAM identities.

This policy provides access to manage IAM Identity Center users from an external provider.

To view the permissions for this policy, see AWSIdentityCenterExternalManagementPolicy 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
AWSIdentityCenterExternalManagementPolicy	Updated managed policy to change the ARN for the provisioning tenant.	December 5, 2025
AWSIdentityCenterExternalManagementPolicy	This policy provides access to manage IAM Identity Center users from an external provider.	November 21, 2025
AWSSSOMasterAccountAdministrator, AWSSSOMemberAccountAdministrator, AWSSSOReadOnly, AWSSSODirectoryAdministrator, AWSSSODirectoryReadOnly	Updated managed policies to include AWS KMS permissions required for IAM Identity Center instances that use customer managed keys for encryption.	September 17, 2025
AWSSSOServiceRolePolicy	This policy now includes permissions to call `identity-sync:DeleteSyncProfile` .	February 11, 2025
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `qapps:ListQAppSessionData` and `qapps:ExportQAppSessionData` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	October 2, 2024

Change	Description	Date
AWSSSOMasterAccountAdministrator	IAM Identity Center added a new action to grant DeleteSyncProfile permissions to allow you to use this policy to delete sync profiles. This is action is associated with DeleteInstance API.	September 26, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `s3:ListCallerAccessGrants` action to support identity-enhanced console sessions for AWS managed applications that support these sessions.	September 4, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `aoss:APIAccessAll`, `es:ESHttpHead`, `es:ESHttpPost`, `es:ESHttpGet`, `es:ESHttpPatch`, `es:ESHttpDelete`, and `es:ESHttpPut` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	July 12, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `qapps:PredictQApp`, `qapps:ImportDocument`, `qapps:AssociateLibraryItemReview`, `qapps:DisassociateLibraryItemReview`, `qapps:GetQAppSession`, `qapps:UpdateQAppSession`, `qapps:GetQAppSessionMetadata`, `qapps:UpdateQAppSessionMetadata`, and `qapps:TagResource` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	June 27, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `elasticmapreduce:AddJobFlowSteps`, `elasticmapreduce:DescribeCluster`, `elasticmapreduce:CancelSteps`, `elasticmapreduce:DescribeStep`, and `elasticmapreduce:ListSteps` actions to support trusted identity propagation in Amazon EMR.	May 17, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `qapps:CreateQApp`, `qapps:PredictProblemStatementFromConversation`, `qapps:PredictQAppFromProblemStatement`, `qapps:CopyQApp`, `qapps:GetQApp`, `qapps:ListQApps`, `qapps:UpdateQApp`, `qapps:DeleteQApp`, `qapps:AssociateQAppWithUser`, `qapps:DisassociateQAppFromUser`, `qapps:ImportDocumentToQApp`, `qapps:ImportDocumentToQAppSession`, `qapps:CreateLibraryItem`, `qapps:GetLibraryItem`, `qapps:UpdateLibraryItem`, `qapps:CreateLibraryItemReview`, `qapps:ListLibraryItems`, `qapps:CreateSubscriptionToken`, `qapps:StartQAppSession`, and `qapps:StopQAppSession` actions to support identity-enhanced console sessions for AWS	April 30, 2024

Change	Description	Date
	managed applications that support these sessions.	
AWSSSOMasterAccountAdministrator	This policy now includes the `signin:CreateTrustedIdentityPropagationApplicationForConsole` and `signin:ListTrustedIdentityPropagationApplicationsForConsole` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 26, 2024
AWSSSOMemberAccountAdministrator	This policy now includes the `signin:CreateTrustedIdentityPropagationApplicationForConsole` and `signin:ListTrustedIdentityPropagationApplicationsForConsole` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 26, 2024

Change	Description	Date
AWSSSOReadOnly	This policy now includes the `signin:ListTrustedIdentityPropagationApplicationsForConsole` action to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 26, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `qbusiness:PutFeedback` action to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 26, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `q:StartConversation`, `q:SendMessage`, `q:ListConversations`, `q:GetConversation`, `q:StartTroubleshootingAnalysis`, `q:GetTroubleshootingResults`, `q:StartTroubleshootingResolutionExplanation`, and `q:UpdateTroubleshootingCommandResult` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 24, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `sts:SetContext` action to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 19, 2024

Change	Description	Date
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `qbusiness:Chat`, `qbusiness:ChatSync`, `qbusiness:ListConversations`, `qbusiness:ListMessages`, and `qbusiness:DeleteConversation` actions to support identity-enhanced console sessions for AWS managed applications that support these sessions.	April 11, 2024
AWSIAMIdentityCenterAllowListForIdentityContext	This policy now includes the `s3:GetAccessGrantsInstanceForPrefix` and `s3:GetDataAccess` actions.	November 26, 2023
AWSIAMIdentityCenterAllowListForIdentityContext	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.	November 15, 2023
AWSSSODirectoryReadOnly	This policy now includes the new namespace `identitystore-auth` with new permissions to allow users to list and get sessions.	February 21, 2023

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 `AWSSSOManageDelegatedAdministrator` 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 `AWSSSOManageDelegatedAdministrator` 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 `PutRolePermisionsBoundary`.	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	This policy now allows all IAM Identity Center Identity Store service actions. For information about the actions available in the IAM Identity Center Identity Store service, see the IAM Identity Center Identity Store API Reference.	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 Understanding service-linked roles in IAM Identity Center.

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 AWSSSOServiceRolePolicy 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 AWSSSOServiceRolePolicy 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 AWSSSOServiceRolePolicy 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 AWSSSOServiceRolePolicy service-linked role permissions policy allows IAM Identity Center to complete the following on all IAM roles (*):

- `iam:listRoles`

The AWSSSOServiceRolePolicy 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 AWSSSOServiceRolePolicy service-linked role permissions policy allows IAM Identity Center to complete the following on “arn:aws:identity-sync:*:*:profile/*”:

- `identity-sync:DeleteSyncProfile`

For more information on updates to the AWSSSOServiceRolePolicy service-linked role permissions policy, see IAM Identity Center updates to AWS managed policies.

JSON

```
{
   "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":[
            "*"
         ]
      },
      {
```

```
            "Sid":"AllowDeleteSyncProfile",
            "Effect":"Allow",
            "Action":[
               "identity-sync:DeleteSyncProfile"
            ],
            "Resource":[
               "arn:aws:identity-sync:*:*:profile/*"
            ]
         }
      ]
}
```

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 do not 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 are 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 do not 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. Remove permission sets in IAM Identity Center 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

Console operation name used before November 15, 2023	API action used after November 15, 2023
DeleteSSO	DeleteInstance
DisassociateProfile	DeleteApplicationAssignment
GetApplicationTemplate	DescribeApplicationProvider
GetManagedApplicationInstance	DescribeApplication
GetSharedSsoConfiguration	DescribeInstance
ListApplicationInstances	ListApplications
ListApplicationTemplates	ListApplicationProviders
ListDirectoryAssociations	DescribeInstance
ListProfileAssociations	ListApplicationAssignments
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

Operation name	API actions used before October 15, 2020	API actions used after October 15, 2020
CreatePermissionSet	CreatePermissionSet	CreatePermissionSet
DeleteApplicationInstanceForAWsAccount	DeleteApplicationInstance | DeleteTrust	DeleteAccountAssignment
DeleteApplicationProfileForAwsAccount	DeleteProfile	DeleteAccountAssignment
DeletePermissionsPolicy	DeletePermissionsPolicy	DeleteInlinePolicyFromPermissionSet
DeletePermissionSet	DeletePermissionSet	DeletePermissionSet
DescribePermissionsPolicies	DescribePermissionsPolicies	ListManagedPoliciesInPermissionSet
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

Operation name	API actions used before October 15, 2020	API actions used after October 15, 2020
ListPermissionSets	ListPermissionSets	ListPermissionSets
ListProfileAssociations	ListProfileAssociations	ListAccountAssignments
ProvisionApplicationInstanceForAWSAccount	GetApplicationInstance | CreateApplicationInstance	CreateAccountAssignment
ProvisionApplicationProfileForAWSAccountInstance	GetProfile | CreateProfile | UpdateProfile	CreateAccountAssignment
ProvisionSAMLProvider	GetTrust | CreateTrust | UpdateTrust	CreateAccountAssignment
PutPermissionsPolicy	PutPermissionsPolicy	PutInlinePolicyToPermissionSet
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

A best practice is to monitor your organization to ensure that changes are logged. Monitoring helps you ensure that you can investigate any unexpected change and roll back unwanted changes. IAM Identity Center currently supports two AWS services that help you monitor your organization and the activity that happens within it: AWS CloudTrail and Amazon EventBridge.

Topics

- Logging IAM Identity Center API calls with AWS CloudTrail
- Logging IAM Identity Center SCIM API calls with AWS CloudTrail
- Connect application components with Amazon EventBridge
- Logging 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 do not 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.

The following table summarizes the CloudTrail events of IAM Identity Center, their CloudTrail event sources, and matching APIs. Refer to the IAM Identity Center API references to learn more about the APIs.

> **Note**
>
> There is an additional group of CloudTrail events, referred to as Sign-in, which AWS emits for signing in to AWS as an IAM Identity Center user. These events have no matching public APIs, and therefore aren't listed in the API references.

CloudTrail events	Public APIs	Description	CloudTrail event sources
IAM Identity Center	IAM Identity Center	The IAM Identity Center APIs enable the managemen t of permission sets, applications, trusted token issuers, account and applicati on assignments, IAM Identity Center instances, and tags.	`sso.amazo naws.com`
Identity Store	Identity Store	The Identity Store APIs enable the management of the life cycle of your workforce's users and groups, and the users' group memberships. Also, they support	`sso-direc tory.amaz onaws.com ,identitys tore.amaz onaws.com`

CloudTrail events	Public APIs	Description	CloudTrail event sources
		the management of users' MFA devices.	
OIDC	OIDC	The OIDC APIs support trusted identity propagati on, and sign-in to AWS CLI and IDE toolkits as an already authenticated IAM Identity Center user.	`sso.amazo naws.com` , `sso-oauth.amazonaw s.com`
AWS access portal	AWS access portal	The AWS access portal APIs support the operations of the AWS access portal and users getting account credentials through the AWS CLI.	`sso.amazo naws.com`
SCIM	SCIM	The SCIM APIs support the provisioning of users, groups, and group membershi ps through the SCIM protocol. See Logging IAM Identity Center SCIM API calls with AWS CloudTrail for more information.	`identitystore-scim.amazonaw s.com`

CloudTrail events	Public APIs	Description	CloudTrail event sources
AWS Sign-In	No public API	AWS emits Sign-in CloudTrail events for user authentication and federation flows into IAM Identity Center.	`signin.amazon.com`

Topics

- CloudTrail use cases for IAM Identity Center
- IAM Identity Center information in CloudTrail

CloudTrail use cases for IAM Identity Center

The CloudTrail events that IAM Identity Center emits can be valuable for a variety of use cases. Organizations can use these event logs to monitor and audit the user access and activity within their AWS environment. This can help compliance use cases, as the logs capture details on who is accessing what resources and when. You can also use the CloudTrail data for incident investigations, allowing teams to analyze user actions and track suspicious behavior. Additionally, the event history can support troubleshooting efforts, providing visibility into changes made to user permissions and configurations over time.

The following sections describe the foundational use cases that inform your workflows such as audit, incident investigation, and troubleshooting.

Identifying the user in IAM Identity Center user-initiated CloudTrail events

IAM Identity Center emits two CloudTrail fields that enable you to identify the IAM Identity Center user behind the CloudTrail events, such as signing into IAM Identity Center or AWS CLI, and using the AWS access portal, including managing MFA devices:

- `userId` – The unique and immutable user identifier from the Identity Store of an IAM Identity Center instance.
- `identityStoreArn` – The Amazon Resource Name (ARN) of the Identity Store that contains the user.

The `userID` and `identityStoreArn` fields display in the `onBehalfOf` element nested inside the `userIdentity` element as shown in the following example CloudTrail event log. This event log shows these two fields on an event where the `userIdentity` type is `"IdentityCenterUser"`. You can also find these fields on events for authenticated IAM Identity Center users where the `userIdentity` type is "Unknown". Your workflows should accept both type values.

```
"userIdentity":{
  "type":"IdentityCenterUser",
  "accountId":"111122223333",
  "onBehalfOf": {
    "userId": "544894e8-80c1-707f-60e3-3ba6510dfac1",
    "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
    },
    "credentialId" : "90e292de-5eb8-446e-9602-90f7c45044f7"
  }
```

Tip

We recommend you use `userId` and `identityStoreArn` for identifying the user behind IAM Identity Center CloudTrail events. The `userName` and `principalId` fields under the `userIdentity` element are no longer available. If your workflows, such as audit or incident response, depend on having access to the `username`, you have two options:

- Retrieve the username from the IAM Identity Center directory as explained in Username in sign-in CloudTrail events.
- Get the `UserName` that IAM Identity Center emits under the `additionalEventData` element in Sign-in. This option doesn't require access to the IAM Identity Center directory. For more information, see Username in sign-in CloudTrail events.

To retrieve the details of a user, including the `username` field, you query the Identity Store with user ID and Identity Store ID as parameters. You can perform this action through the `DescribeUser` API request or through the CLI. The following is an example CLI command. You can omit the `region` parameter if your IAM Identity Center instance is in the CLI default Region.

```
aws identitystore describe-user \
--identity-store-id  d-1234567890 \
--user-id  544894e8-80c1-707f-60e3-3ba6510dfac1 \
```

```
--region your-region-id
```

To determine the Identity Store ID value for the CLI command in the previous example, you can extract the Identity Store ID from the `identityStoreArn` value. In the example ARN `arn:aws:identitystore::111122223333:identitystore/d-1234567890`, the Identity Store ID is `d-1234567890`. Alternatively, you can locate the Identity Store ID by navigating to **Identity Store** tab in the **Settings** section of the IAM Identity Center console.

If you are automating the lookup of users in the IAM Identity Center directory, we recommend that you estimate the frequency of user lookups, and consider the IAM Identity Center throttle limit on the Identity Store API. Caching retrieved user attributes can help you stay within the throttle limit.

Correlating user events within the same user session

The `AuthWorkflowID` field emitted in sign-in events enables tracking all CloudTrail events associated with a sign-in sequence before the commencement of an IAM Identity Center user session.

For user actions inside the AWS access portal, the `credentialId` value is set to the ID of the IAM Identity Center user's session used to request the action. You can use this value to identify CloudTrail events initiated within the same authenticated IAM Identity Center user session in the AWS access portal.

> **Note**
>
> You can't use `credentialId` to correlate sign-in events to the subsequent events, such as the use of the AWS access portal. The value of the `credentialId` field emitted in sign-in events has internal use, and we recommend that you not rely on it. The value of the `credentialId` field emitted for AWS access portal events invoked with OIDC equals the ID of the access token.

Identifying user background session details in IAM Identity Center user-initiated CloudTrail events

The following CloudTrail event captures the process of OAuth 2.0 token exchange, in which an existing access token (the `subjectToken`) that represents the user's interactive session is exchanged for a refresh token (the `requestedTokenType`). The refresh token allows any user

initiated long-running jobs to continue running with the user's permissions, even after the user signs out.

For IAM Identity Center user background sessions, the CloudTrail event includes an additional element called `resource` in the `requestParameters` element. The `resource` parameter contains the Amazon Resource Name (ARN) of the job that runs in the background. This element is present only in CloudTrail event records and is not included in the IAM Identity Center CreateTokenWithIAM API or SDK responses.

```
{
  "clientId": "EXAMPLE-CLIENT-ID",
  "grantType": "urn:ietf:params:oauth:grant-type:token-exchange",
  "code": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "redirectUri": "https://example.com/callback",
  "assertion": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "subjectToken": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "subjectTokenType": "urn:ietf:params:oauth:token-type:access_token",
  "requestedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",
  "resource": "arn:aws:sagemaker:us-west-2:123456789012:training-job/my-job"
}
```

Correlating users between IAM Identity Center and external directories

IAM Identity Center provides two user attributes that you can use to correlate a user in its directory to the same user in an external directory (for example, Microsoft Active Directory and Okta Universal Directory).

- `externalId` – The external identifier of an IAM Identity Center user. We recommend you map this identifier to an immutable user identifier in the external directory. Note that IAM Identity Center doesn't emit this value in CloudTrail.
- `username` – A customer-provided value that users usually sign in with. The value can change (for example, with a SCIM update). Note that when the identity source is Directory Service, the username that IAM Identity Center emits in CloudTrail matches the username that you enter to authenticate. The username doesn't need to be an exact match to the username in the IAM Identity Center directory.

  If you have access to the CloudTrail events but not the IAM Identity Center directory, you can use the username emitted under the `additionalEventData` element at sign-in. For more details about username in `additionalEventData`, refer to Username in sign-in CloudTrail events.

The mapping of these two user attributes to corresponding user attributes in an external directory is defined in IAM Identity Center when the identity source is the Directory Service. For infomration, see Attribute mappings between IAM Identity Center and External Identity Providers directory. External IdPs that provision users with SCIM have their own mapping. Even if you use the IAM Identity Center directory as the identity source, you can use the `externalId` attribute to cross-reference security principals to your external directory.

The following section explains how you can look up an IAM Identity Center user given the user's `username` and `externalId`.

Viewing an IAM Identity Center user by username and externalId

You can retrieve user attributes from the IAM Identity Center directory for a known username by first requesting a corresponding `userId` using the `GetUserId` API request, then issue a `DescribeUser` API request, as shown in the previous example. The following example demonstrates how you can retrieve a `userId` from the Identity Store for a specific username. You can omit the `region` parameter if your IAM Identity Center instance is in the default Region with the CLI.

```
aws identitystore get-user-id \
    --identity-store d-9876543210 \
    --alternate-identifier '{
      "UniqueAttribute": {
      "AttributePath": "username",
      "AttributeValue": "anyuser@example.com"
        }
          }' \
    --region your-region-id
```

Similarly, you can use the same mechanism when you know the `externalId`. Update the attribute path in the previous example with the `externalId` value, and the attribute value with the specific `externalId` for which you are searching.

Viewing a user's Secure Identifier (SID) in Microsoft Active Directory (AD) and externalId

In certain cases, IAM Identity Center emits a user's SID in the `principalId` field of CloudTrail events, such as those that the AWS access portal and OIDC APIs emit. **These cases are being phased out.** We recommend your workflows use the AD attribute `objectguid` when you need a unique user identifier from AD. You can find this value in the `externalId` attribute in the IAM

Identity Center directory. However, if your workflows require the use of SID, retrieve the value from AD as it's not available through IAM Identity Center APIs.

Correlating user events within the same user session describes how you can use the `externalId` and `username` fields to correlate an IAM Identity Center user to a matching user in an external directory. By default, IAM Identity Center maps `externalId` to the `objectguid` attribute in AD, and this mapping is fixed. IAM Identity Center allows administrators the flexibility to map `username` differently than its default mapping to `userprincipalname` in AD.

You can view these mappings in the IAM Identity Center console. Navigate to the **Identity Source** tab of **Settings**, and choose **Manage sync** in the **Actions** menu. In the **Manage Sync** section, choose the **View attribute mappings** button.

While you can use any unique AD user identifier available in IAM Identity Center to look up a user in AD, we recommend using the `objectguid` in your queries because it is an immutable identifier. The following example shows how to query Microsoft AD with Powershell to retrieve a user using the user's `objectguid` value of `16809ecc-7225-4c20-ad98-30094aefdbca`. A successful response to this query includes the user's SID.

```
Install-WindowsFeature -Name  RSAT-AD-PowerShell

  Get-ADUser `
  -Filter {objectGUID -eq  [GUID]::Parse("16809ecc-7225-4c20-ad98-30094aefdbca")} `
  -Properties *
```

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.

> **Note**
>
> For more information about how user identification and tracking of user actions in CloudTrail events is evolving, refer to Important changes to CloudTrail events for IAM Identity Center in the *AWS Security Blog*.

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 topics in the *AWS CloudTrail User Guide*:

- 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.

CloudTrail events for supported IAM Identity Center APIs

The following sections provide information about the CloudTrail events associated with the following APIs that IAM Identity Center supports:

- IAM Identity Center API
- Identity Store API
- OIDC API
- AWS access portal API
- SCIM API

CloudTrail events of IAM Identity Center API operations

The following list contains the CloudTrail events that the public IAM Identity Center operations emit with the `sso.amazonaws.com` event source. For more information about the public IAM Identity Center API operations, see the IAM Identity Center API Reference.

You might find additional events in CloudTrail for IAM Identity Center console API operations that the console relies on. For more information about these console APIs, see the Service Authorization Reference.

- AttachCustomerManagedPolicyReferenceToPermissionSet
- AttachManagedPolicyToPermissionSet
- CreateAccountAssignment
- CreateApplication
- CreateApplicationAssignment
- CreateInstance
- CreateInstanceAccessControlAttributeConfiguration
- CreatePermissionSet
- CreateTrustedTokenIssuer
- DeleteAccountAssignment
- DeleteApplication
- DeleteApplicationAccessScope
- DeleteApplicationAssignment
- DeleteApplicationAuthenticationMethod
- DeleteApplicationGrant
- DeleteInlinePolicyFromPermissionSet
-

DeleteInstance

- DeleteInstanceAccessControlAttributeConfiguration
- DeletePermissionsBoundaryFromPermissionSet
- DeletePermissionSet
- DeleteTrustedTokenIssuer
- DescribeAccountAssignmentCreationStatus
s
- DescribeAccountAssignmentDeletionStatus
- DescribeApplication
- DescribeApplicationAssignment
- DescribeApplicationProvider
- DescribeInstance
- DescribeInstanceAccessControlAttributeConfiguration
- DescribePermissionSet
- DescribePermissionSetProvisioningStatus
- DescribeTrustedTokenIssuer
-

DetachCustomerManagedPolicyReferenceFromPermissionSet

- DetachManagedPolicyFromPermissionSet
- GetApplicationAccessScope
- GetApplicationAssignmentConfiguration
- GetApplicationAuthenticationMethod
- GetApplicationGrant
- GetInlinePolicyForPermissionSet
- GetPermissionsBoundaryForPermissionSet
- ListAccountAssignmentCreationStatus
- ListAccountAssignmentDeletionStatus
- ListAccountAssignments
- ListAccountAssignmentsForPrincipal
- ListAccountsForProvisionedPermissionSet
- ListApplicationAccessScopes
- ListApplicationAssignments
- ListApplicationAssignmentsForPrincipal

- ListApplicationAuthenticationMethods
- ListApplicationGrants
- ListApplicationProviders
- ListApplications
- ListCustomerManagedPolicyReferencesInPermissionSet
- ListInstances
- ListManagedPoliciesInPermissionSet
- ListPermissionSetProvisioningStatus
- ListPermissionSets
- ListPermissionSetsProvisionedToAccount
- ListTagsForResource
- ListTrustedTokenIssuers
- ProvisionPermissionSet
- PutApplicationAccessScope
- PutApplicationAssignmentConfiguration

- PutApplicationAuthenticationMethod
- PutApplicationGrant
- PutInlinePolicyToPermissionSet
- PutPermissionsBoundaryToPermissionSet
- TagResource
- UntagResource
- UpdateApplication
- UpdateInstance
- UpdateInstanceAccessControlAttributeConfiguration
- UpdatePermissionSet
- UpdateTrustedTokenIssuer

CloudTrail events of Identity Store API operations

The following list contains the CloudTrail events that the public Identity Store operations emit with the `identitystore.amazonaws.com` event source. For more information about the public Identity Store API operations, see the Identity Store API Reference.

You might find additional events in CloudTrail for the Identity Store console API operations with the `sso-directory.amazonaws.com` event source. These APIs support the console and AWS access portal. If you need to detect the occurrence of a particular operation, such as adding member to a group, we recommend you consider both public and console API operations. For more information about these console APIs, see the Service Authorization Reference.

- CreateGroup
- CreateGroupMembership
- CreateUser
- DeleteGroup
- DeleteGroupMembership
- DeleteUser
- DescribeGroup
- DescribeGroupMembership
- DescribeUser
- GetGroupId
- GetGroupMembershipId
- GetUserId
- IsMemberInGroups
- ListGroupMemberships
- ListGroupMembershipsForMember
- ListGroups
- ListUsers
- UpdateGroup
- UpdateUser

CloudTrail events of OIDC API operations

The following list contains the CloudTrail events that the public OIDC operations emit. For more information about the public OIDC API operations, see the OIDC API Reference.

- CreateToken (event source `sso.amazonaws.com`)
- CreateTokenWithIAM (event source `sso-oauth.amazonaws.com`)

CloudTrail events of AWS access portal API operations

The following list contains the CloudTrail events that the AWS access portal API operations emit with the `sso.amazonaws.com` event source. The API operations noted as being unavailable in

the public API support the operations of the AWS access portal. Using the AWS CLI can lead to the emission of CloudTrail events of both the public AWS access portal API operations and those that are unavailable in the public API. For more information about public AWS access portal API operations, see the AWS access portal API Reference.

- Authenticate (Not available in the public API. Provides login to the AWS access portal.)
- Federate (Not available in the public API. Provides federation into applications.)
- ListAccountRoles
- ListAccounts
- ListApplications (Not available in the public API. Provides users' assigned resources for display in the AWS access portal.)
- ListProfilesForApplication (Not available in the public API. Provides application metadata for display in the AWS access portal.)
- GetRoleCredentials
- Logout

CloudTrail events of SCIM API operations

For information about public SCIM API operations, see AWS access portal API Reference.

Identity information in IAM Identity Center CloudTrail events

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.
- Whether the request was made by an IAM Identity Center user. If so, the `userId` and `identityStoreArn` fields are available in the CloudTrail events to identify the IAM Identity Center user who initiated the request. For more information, see Identifying the user in IAM Identity Center user-initiated CloudTrail events.

For more information, see the CloudTrail userIdentity element.

> **Note**
>
> Currently, IAM Identity Center doesn't emit CloudTrail events for user sign-in to AWS managed web applications (for example, Amazon SageMaker AI Studio) with the OIDC API. These web applications are a subset of the broader set of the section called "AWS managed applications", which also include non-web applications such as Amazon Athena SQL and Amazon S3 Access Grants.

Understanding CloudTrail events for IAM Identity Center

A trail is a configuration that enables delivery of events to an Amazon S3 bucket that you specify. 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 events aren't an ordered stack trace of the public API calls, so they do not appear in any specific order. Learn about the contents of a CloudTrail record in the *CloudTrail User Guide*.

This example demonstrates a CloudTrail log entry capturing a `DescribePermissionsPolicies` action performed by an IAM user (samadams) interacting with IAM Identity Center:

```
{
   "Records":[
      {
         "eventVersion":"1.05",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDAJAIENLMexample",
            "arn":"arn:aws:iam::08966example:user/samadams",
            "accountId":"111122223333",
            "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":"111122223333"
      }
   ]
}
```

This example demonstrates a CloudTrail log entry capturing a `ListApplications` action performed by an IAM Identity Center user within the AWS access portal:

```
{
   "Records":[
      {
         "eventVersion":"1.05",
         "userIdentity":{
            "type":"IdentityCenterUser",
            "accountId":"111122223333",
            "onBehalfOf": {
              "userId": "94d00cd8-e9e6-4810-b177-b08e84775435",
              "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
            },
            "credentialId" : "cdee2490-82ed-43b3-96ee-b75fbf0b97a5"
         },
         "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":"111122223333"
      }
   ]
}
```

This example demonstrates a CloudTrail log entry capturing a `CreateToken` action performed by an IAM Identity Center user authenticating with the IAM Identity Center OIDC service:

```
{
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IdentityCenterUser",
        "accountId": "111122223333",
        "onBehalfOf": {
          "userId": "94d00cd8-e9e6-4810-b177-b08e84775435",
          "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
        },
        "credentialId" : "cdee2490-82ed-43b3-96ee-b75fbf0b97a5"
      },
      "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": "111122223333",
          "type": "IdentityStoreId",
          "ARN": "d-1234567890"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
}
```

Understanding IAM Identity Center sign-in events

AWS CloudTrail records successful and unsuccessful sign-in events for all IAM Identity Center identity sources. IAM Identity Center and Active Directory (AD Connector and AWS Managed Microsoft AD) sourced identities include additional sign-in events that are captured each time a user is prompted to solve a specific credential challenge or factor, in addition to 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
`CredentialChallenge`	Used to notify that IAM Identity Center has requested the user to solve a specific credential challenge and specifies the `CredentialType` that was required (For example, PASSWORD or TOTP).	Native IAM Identity Center users, AD Connector, and AWS Managed Microsoft AD
`CredentialVerification`	Used to notify that the user has attempted to solve a specific `CredentialChallenge` 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. *Users failing to successfully complete the required credential challenges will result in no `UserAuthentication` event being logged.*	All identity sources

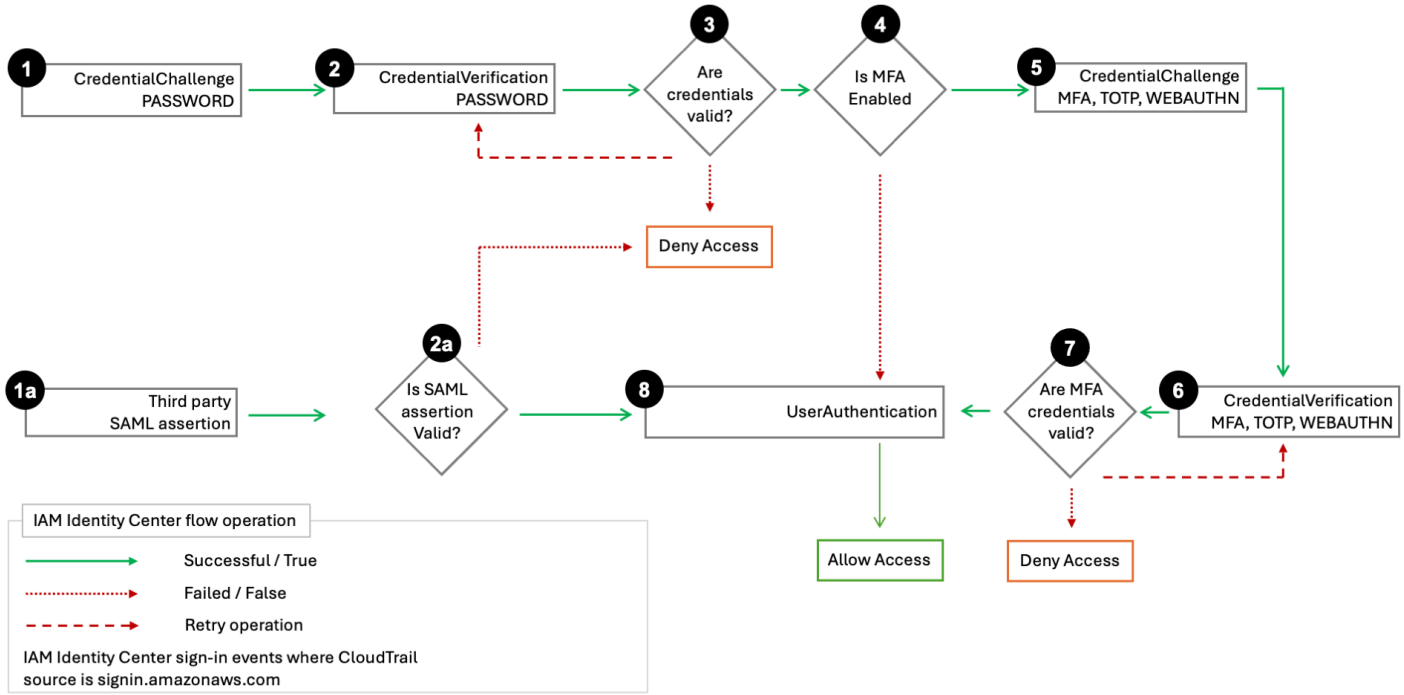
The following table captures additional useful event data fields contained within specific sign-in CloudTrail events.

Field	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. `UserAuthe`	`CredentialChallenge`, `CredentialVerifica`	CredentialType": "PASSWORD" or "CredentialType": "PASSWORD

Field	Event purpose	Sign-in event applicability	Example values
	`ntication` events will include all of the `CredentialType` values that were successfully verified across the user's sign-in sequence.	`tion` , `UserAuthentication`	,TOTP" (possible values include: PASSWORD, TOTP, WEBAUTHN, EXTERNAL_IDP, RESYNC_TOTP, EMAIL_OTP)
`DeviceEnrollmentRequired`	Used to specify that the user was required to register an MFA device during sign-in, and that the user successfully completed that request.	`UserAuthentication`	"DeviceEnrollmentRequired": "true"
`LoginTo`	Used to specify the redirect location following a successful sign-in sequence.	`UserAuthentication`	"LoginTo": "https://mydirectory.awsapps.com/start/....."

CloudTrail events in the IAM Identity Center sign-in flows

The following diagram describes the sign-in flow and the CloudTrail events that Sign-in emits.



The diagram shows a **password sign-in** flow and a **federated sign-in** flow.

The **password sign-in** flow, which consists of steps 1–8, demonstrates the steps during the username and password sign-in process. IAM Identity Center sets `userIdentity.additionalEventData.CredentialType` to "`PASSWORD`", and IAM Identity Center goes through the credentials challenge-response cycle, retrying as needed.

The number of steps depends on the type of login and the presence of the multi-factor authentication (MFA). The initial process results in three or five CloudTrail events with `UserAuthentication` ending the sequence for a successful authentication. Unsuccessful password authentication attempts result in additional CloudTrail events as the IAM Identity Center re-issues `CredentialChallenge` for regular or, if enabled, MFA authentication.

The password sign-in flow also covers the scenario where an IAM Identity Center user newly-created with a `CreateUser` API call signs in with a one-time password (OTP). The credential type in this scenario is “`EMAIL_OTP`”.

The **federated sign-in** flow, consisting of steps 1a, 2a, and 8, demonstrates the main steps during the federated authentication process where a SAML assertion is provided by an identity provider, validated by IAM Identity Center, and if successful, results in `UserAuthentication`. IAM Identity

Center doesn't invoke the internal MFA authentication sequence in steps 3 – 7 because an external, federated identity provider is responsible for all user credential authentication.

Username in sign-in CloudTrail events

IAM Identity Center emits the `UserName` field under the `additionalEventData` element once per successful sign-in of an IAM Identity Center user. The following list describes the two sign-in events in scope, and the conditions under which these events happen. Only one of the conditions can be true when a user is signing in.

- `CredentialChallenge`
  - When `CredentialType` is "`PASSWORD`" – applies to password authentication with Directory Service or IAM Identity Center directory.
  - When `CredentialType` is "`EMAIL_OTP`" – applies only to the IAM Identity Center directory when a user created with a `CreateUser` API call attempts to sign in for the first time, and the user receives a one-time password to sign in with that password once.
- `UserAuthentication`
  - When `CredentialType` is "`EXTERNAL_IDP`" – applies to authentication with an external IdP.

The value of `UserName` for successful authentications is as follows :

- When the identity source is an external IdP, the value is equal to the `nameID` value in the incoming SAML assertion. This value is equal to the `UserName` field in the IAM Identity Center directory.
- When the identity source is an IAM Identity Center directory, the value emitted is equal to the `UserName` field in this directory.
- When the identity source is the Directory Service, the value emitted is equal to the username that the user enters during authentication. For example, a user who has the username `anyuser@company.com`, can authenticate with `anyuser`, `anyuser@company.com`, or `company.com/anyuser`, and in each case the entered value is emitted in CloudTrail respectively.

Security masking of incorrect username attempts

The `UserName` field contains the string `HIDDEN_DUE_TO_SECURITY_REASONS` when the recorded event is a console sign-in failure caused by incorrect user name input. CloudTrail doesn't record

the contents in this case because the text could contain sensitive information, as described in the following examples:

- A user accidentally types a password in the user name field.
- A user accidentally types the account name of a personal email account, a bank sign-in identifier, or some other private ID.

> **Tip**
>
> We recommend you use `userId` and `identityStoreArn` for identifying the user behind IAM Identity Center CloudTrail events. If you need to use the `userName` field, you can use the `userName` under the `additionalEventData` element that's emitted once per successful sign-in.

For additional information on how you can use the `UserName` field, refer to Correlating user events within the same user session.

Example events for IAM Identity Center sign-in scenarios

The following examples illustrate the typical CloudTrail event sequences generated during various AWS sign-in scenarios. These examples serve as reference patterns to help you interpret authentication logs, identify security issues, and verify that your authentication policies are functioning correctly.

Topics

- Successful sign-in when authenticating with password only
- Successful sign-in when authenticating with an external identity provider
- Successful sign-in when authenticating with a password and a time-based one-time password (TOTP) authenticator app
- Successful sign-in when authenticating with a password and forced MFA registration is required
- Failed sign-in due to incorrect password authentication

Successful sign-in when authenticating with password only

The following sequence of events captures an example of a successful password only sign-in.

CredentialChallenge (Password)


```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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",
      "UserName":"bobsmith@example.com",
      "CredentialType":"PASSWORD"
   },
   "requestID":"5be44ffb-6946-4f47-acaf-1adebd4afead",
   "eventID":"27ea7725-c1fd-4355-bdba-d0e628e0e604",
   "readOnly":false,
   "eventType":"AwsServiceEvent",
   "managementEvent":true,
   "eventCategory":"Management",
   "serviceEventDetails":{
      "CredentialChallenge":"Success"
   }
}
```


Successful CredentialVerification (Password)

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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
BshlIc5OBAA6ftz73M6LsfLWDlfOxviO2K3wet946lC30f_iWdilx-
zv__4pSHf7mcUIs&wdc_csrf_token=srAzW1jK4GPYYoR452ruZ38DxEsDY9x81q1tVRSnno5pUjISvP7TqziOLiBLBUSx
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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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
BshlIc5OBAA6ftz73M6LsfLWDlfOxviO2K3wet946lC30f_iWdilx-
zv__4pSHf7mcUIs&wdc_csrf_token=srAzW1jK4GPYYoR452ruZ38DxEsDY9x81q1tVRSnno5pUjISvP7TqziOLiBLBUSx
east-1",
      "CredentialType":"EXTERNAL_IDP",
      "UserName":"bobsmith@example.com"
   },
   "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 time-based one-time password (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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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",
```

```
      "UserName":"bobsmith@example.com"
   },
   "requestID":"e454ea66-1027-4d00-9912-09c0589649e1",
   "eventID":"d89cc0b5-a23a-4b88-843a-89329aeaef2e",
   "readOnly":false,
   "eventType":"AwsServiceEvent",
   "managementEvent":true,
   "eventCategory":"Management",
   "recipientAccountId":"111122223333",
   "serviceEventDetails":{
      "CredentialChallenge":"Success"
   }
}
```

Successful CredentialVerification (Password)

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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
\u003dQVlBQmVLeFhWeDRmZFJmMmxHcWYwdzhZck5RQUlnQUJBQk5FWVhSaFVHeGhibVZUZEdGMFpWQmhjbUZ0QUFsUVpYS
\u0026auth_code
\u003d11Fir1mCVJ-4Y5UY6RI10UCXvRePCHd6195xvYg1rwo1Pj7B-7UGIGlYUUVe31Nkzd7ihxKn6DMdnFfO01O8qc3RF
Sx-pjBXKG_jUcvBk_UILdGytV4o1u97h42B-
TA_6uwdmJiw1dcCz_Rv44d_BS0PkulW-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 demonstrates a successful password authentication where the user was required to register and successfully complete multi-factor authentication (MFA) before finalizing their sign-in process.

CredentialChallenge (Password)

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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",
      "UserName":"bobsmith@example.com"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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":"IdentityCenterUser",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
      "onBehalfOf": {
         "userId": "94d00cd8-e9e6-4810-b177-b08e84725435",
         "identityStoreArn": "arn:aws:identitystore::111122223333:identitystore/
d-1234567890"
      },
      "credentialId" : "8f761cae-883d-4a3d-af67-3abf46488f71"
   },
   "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_maUsZ7ABETjeQhyWfvIHYz52rgR28sYAKN1oEk2G07czrwzXvE9HLlN2K9De8LyBEV83SFeDQfrWpkwXf
FJyJqkoGrt_w6rm_MpAn0uyrVq8udY EgU3fhOL3QWvWiquYnDPMyPmmy_qkZgR9rz__BI
\u0026wdc_csrf_token
\u003dJih9U62o5LQDtYLNqCK8a6xj0gJg5BRWq2tbl75y8vAmwZhAqrgrgbxXat2M646UZGp93krw7WYQdHIgi5OYI9QSc
\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 due to incorrect password authentication

The following sequence of events demonstrates an authentication attempt where the user successfully entered their username but failed the password verification step, resulting in an unsuccessful sign-in.

CredentialChallenge (Password)


```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"Unknown",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
   },
   "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",
      "UserName":"bobsmith@example.com"
   },
   "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",
      "arn":"",
      "accountId":"111122223333",
      "accessKeyId":"",
   },
   "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"
    }
}
```

Logging IAM Identity Center SCIM API calls with AWS CloudTrail

IAM Identity Center SCIM is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures API calls for SCIM as events. Using the information collected by CloudTrail, you can determine the information about the requested action, the date and time of the action, request parameters, and so on. To learn more about CloudTrail, see AWS CloudTrail User Guide.

> **Note**
>
> CloudTrail is enabled on your AWS account when you create the account. However, you might need to rotate your access token to see events from SCIM, if your token was created prior to September 2024.
>
> For more information, see Rotate an access token.

SCIM supports logging for the following operations as events in CloudTrail:

- CreateGroup
- CreateUser
- DeleteGroup
- DeleteUser
- GetGroup
- GetSchema

- GetUser
- ListGroups
- ListResourceTypes
- ListSchemas
- ListUsers
- PatchGroup
- PatchUser
- PutUser
- ServiceProviderConfig

Example CloudTrail events

The following examples demonstrate typical CloudTrail event logs generated during SCIM operations with IAM Identity Center. These examples show the structure and content of events for successful operations and common error scenarios, helping you understand how to interpret CloudTrail logs when troubleshooting SCIM provisioning issues.

Successful `CreateUser` operation

This CloudTrail event shows a successful `CreateUser` operation performed through the SCIM API. The event captures both the request parameters (with sensitive information masked) and the response elements, including the newly-created user's ID. This type of event is generated when an identity provider successfully provisions a new user to IAM Identity Center using the SCIM protocol.

```
{
  "eventVersion": "1.10",
  "userIdentity": {
    "type": "WebIdentityUser",
    "accountId": "123456789012",
    "accessKeyId": "xxxx"
  },
  "eventTime": "xxxx",
  "eventSource": "identitystore-scim.amazonaws.com",
  "eventName": "CreateUser",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "xx.xxx.xxx.xxx",
  "userAgent": "Go-http-client/2.0",
  "requestParameters": {
```

```
    "httpBody": {
      "displayName": "HIDDEN_DUE_TO_SECURITY_REASONS",
      "schemas" : [
        "urn:ietf:params:scim:schemas:core:2.0:User"
      ],
      "name": {
        "familyName": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "givenName": "HIDDEN_DUE_TO_SECURITY_REASONS"
      },
      "active": true,
      "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "tenantId": "xxxx"
  },
  "responseElements": {
    "meta" : {
      "created" : "Oct 10, 2024, 1:23:45 PM",
      "lastModified" : "Oct 10, 2024, 1:23:45 PM",
      "resourceType" : "User"
    },
    "displayName" : "HIDDEN_DUE_TO_SECURITY_REASONS",
    "schemas" : [
      "urn:ietf:params:scim:schemas:core:2.0:User"
    ],
    "name": {
      "familyName": "HIDDEN_DUE_TO_SECURITY_REASONS",
      "givenName": "HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "active": true,
    "id" : "c4488478-a0e1-700e-3d75-96c6bb641596",
    "userName": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "requestID": "xxxx",
  "eventID": "xxxx",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "clientProvidedHostHeader": "scim.us-east-1.amazonaws.com"
  }
}
```

Failed `PatchGroup` operation: Missing required path attribute

This CloudTrail event shows a failed `PatchGroup` operation that resulted in a `ValidationException` with the error message `"Missing path in PATCH request"`. The error occurred because the `PATCH` operation requires a path attribute to specify which group attribute to modify, but this attribute was missing from the request.


```
{
  "eventVersion": "1.10",
  "userIdentity": {
    "type": "Unknown",
    "accountId": "123456789012",
    "accessKeyId": "xxxx"
  },
  "eventTime": "xxxx",
  "eventSource": "identitystore-scim.amazonaws.com",
  "eventName": "PatchGroup",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "xxx.xxx.xxx.xxx",
  "userAgent": "Go-http-client/2.0",
  "errorCode": "ValidationException",
  "errorMessage": "Missing path in PATCH request",
  "requestParameters": {
    "httpBody": {
      "operations": [
        {
          "op": "REMOVE",
          "value": "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
      ],
      "schemas": [
        "HIDDEN_DUE_TO_SECURITY_REASONS"
      ]
    },
    "tenantId": "xxxx",
    "id": "xxxx"
  },
  "responseElements": null,
  "requestID": "xxxx",
  "eventID": "xxxx",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
```

```
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "clientProvidedHostHeader": "scim.us-east-1.amazonaws.com"
  }
}
```

Failed `CreateGroup` operation: Group name already exists

This CloudTrail event shows a failed `CreateGroup` operation that resulted in a `ConflictException` with the error message `"Duplicate GroupDisplayName"`. This error occurs when attempting to create a group with a display name that already exists in IAM Identity Center. The identity provider must use a unique group name or update the existing group instead of creating a new one.

```
{
  "eventVersion": "1.10",
  "userIdentity": {
    "type": "Unknown",
    "accountId": "123456789012",
    "accessKeyId": "xxxx"
  },
  "eventTime": "xxxx",
  "eventSource": "identitystore-scim.amazonaws.com",
  "eventName": "CreateGroup",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "xxx.xxx.xxx.xxx",
  "userAgent": "Go-http-client/2.0",
  "errorCode": "ConflictException",
  "errorMessage": "Duplicate GroupDisplayName",
  "requestParameters": {
    "httpBody": {
      "displayName": "HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "tenantId": "xxxx"
  },
  "responseElements": null,
  "requestID": "xxxx",
  "eventID": "xxxx",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
```

```
  "eventCategory": "Management",
  "tlsDetails": {
    "clientProvidedHostHeader": "scim.us-east-1.amazonaws.com"
  }
}
```

Failed `PatchUser` operation: Multiple email addresses not supported

This CloudTrail event shows a failed `PatchUser` operation that resulted in a `ValidationException` with the error message `"List attribute emails exceeds allowed limit of 1"`. This error occurs when attempting to assign multiple email addresses to a user, as IAM Identity Center supports only one email address per user. The identity provider must configure SCIM mapping to send only a single email address for each user.

```
{
  "eventVersion": "1.10",
  "userIdentity": {
    "type": "Unknown",
    "accountId": "123456789012",
    "accessKeyId": "xxxx"
  },
  "eventTime": "xxxx",
  "eventSource": "identitystore-scim.amazonaws.com",
  "eventName": "PatchUser",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "xxx.xxx.xxx.xxx",
  "userAgent": "Go-http-client/2.0",
  "errorCode": "ValidationException",
  "errorMessage": "List attribute emails exceeds allowed limit of 1",
  "requestParameters": {
    "httpBody": {
      "operations": [
        {
          "op": "REPLACE",
          "path": "emails",
          "value": "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
      ],
      "schemas": [
        "HIDDEN_DUE_TO_SECURITY_REASONS"
      ]
    },
    "tenantId": "xxxx",
```

```
    "id": "xxxx"
  },
  "responseElements": null,
  "requestID": "xxxx",
  "eventID": "xxxx",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "clientProvidedHostHeader": "scim.us-east-1.amazonaws.com"
  }
}
```

Common SCIM API validation errors in IAM Identity Center

The following validation error messages commonly appear in CloudTrail events when using the SCIM API with IAM Identity Center. These validation errors typically occur during user and group provisioning operations.

For detailed guidance on resolving these errors and properly configuring SCIM provisioning, see this AWS re:Post article.

- List attribute email exceeds allowed limit of 1
- List attribute addresses allowed limit of 1
- 1 validation errors detected: Value at '*name.familyName*' failed to satisfy constraint: Member must satisfy regular expression pattern: [\\p{L}\\p{M}\\p{S}\\p{N}\\p{P}\\t\\n\\r]+
- 2 validation errors detected: Value at 'name.familyName' failed to satisfy constraint: Member must have length greater than or equal to 1; Value at 'name.familyName' failed to satisfy constraint: Member must satisfy regular expression pattern: [\\p{L}\\p{M}\\p{S}\\p{N}\\p{P}\\t\\n\\r]+
- 2 validation errors detected: Value at 'urn:ietf:params:scim:schemas:extension:enterprise:2.0:User.manager.value' failed to satisfy constraint: Member must have length greater than or equal to 1; Value at 'urn:ietf:params:scim:schemas:extension:enterprise:2.0:User.manager.value' failed to satisfy constraint: Member must satisfy regular expression pattern: [\\p{L}\\p{M}\\p{S}\\p{N}\\p{P}\\t\\n\\r]+",
- Invalid JSON from RequestBody

- Invalid Filter format

Connect application components with Amazon EventBridge

You can integrate IAM Identity Center with Amazon EventBridge to raise events that initiate administrative notifications or invoke automated workflows in response to specific IAM Identity Center actions recorded in CloudTrail events.

For example, you might configure EventBridge rules to detect when a user deletes an application or when IAM Identity Center creates a new group. Depending on your use case, you can route these events to an Amazon SNS topic to notify administrators or invoke additional automation using AWS Lambda, Step Functions, or other EventBridge-supported services.

Logging configurable AD sync errors

You can enable logging on your configurable Active Directory (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 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 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 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**.

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 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.

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. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

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.

Data protection in IAM Identity Center

The AWS Shared Responsibility Model applies to data protection in AWS IAM Identity Center. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

We recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with IAM Identity Center.
- Use TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS IAM Identity Center, or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for diagnostic logs.

Encryption in transit

IAM Identity Center protects data in transit, as it travels to and from the service, by automatically encrypting all inter-network data using the Transport Layer Security (TLS) 1.2 or TLS 1.3 encryption protocol. Direct HTTPS requests authenticated with IAM and sent to the IAM Identity Center APIs, Identity Store API, or OIDC API, are signed using the AWS Signature Version 4 Algorithm to establish a secure connection.

Data privacy

With IAM Identity Center, you retain control of your organization's data. Your user and group identities stored in IAM Identity Center are shared with other AWS services such as AWS managed applications only if you enable them with IAM Identity Center, and if needed by those services.

For additional information, see the AWS Data Privacy FAQ.

Data retention

IAM Identity Center stores your data such as user and group identities, and metadata, until you delete them from the service. When you delete an IAM Identity Center instance, the data it contains is also deleted.

Encryption at rest

IAM Identity Center provides encryption to protect customer data at rest using the following key types:

- **AWS owned keys (default key type)** — IAM Identity Center uses these keys by default to automatically encrypt your data. You can't view, manage, audit their use, or use AWS owned keys for other purposes. IAM Identity Center handles the key management entirely to keep your data secure, without your having to take any action. For more information, see AWS owned keys in the *AWS Key Management Service Developer Guide*.
- **Customer managed keys** — In organization instances of IAM Identity Center, you can choose a symmetric customer managed key for encryption at rest of your workforce identity data such as user and group attributes. You create, own, and manage these encryption keys. Because you have full control of this layer of encryption, you can perform such tasks as:
  - Establishing and maintaining key policies to restrict access to the key to only IAM principals who need access, such as IAM Identity Center and AWS managed applications in the same AWS Organizations and their administrators.
  - Establishing and maintaining IAM policies for access to the key including cross-account access
  - Enabling and disabling key policies
  - Rotating key cryptographic material
  - Auditing access to your data that requires key access
  - Adding tags
  - Creating key aliases
  - Scheduling keys for deletion

To learn how to implement a customer managed KMS key in IAM Identity Center see Implementing customer managed KMS keys in AWS IAM Identity Center. For more information about customer managed keys, see customer managed key in the *AWS Key Management Service Developer Guide*.

> **Note**
>
> IAM Identity Center automatically enables encryption at rest using AWS owned KMS keys to protect customer data at no charge. However, AWS KMS charges apply when using a customer managed key. For more information about pricing, see the AWS Key Management Service pricing.

Considerations for implementing customer managed keys:

- **Dedicated keys**: We recommend creating a new dedicated customer managed KMS key for each IAM Identity Center instance rather than reusing an existing key. This approach provides clearer separation of duties, simplifies access control management, and makes security auditing more straightforward. Having a dedicated key also reduces risk by limiting the impact of key changes to a single IAM Identity Center instance.
- **Use of IAM Identity Center across multiple AWS Regions**: If you plan to replicate your IAM Identity Center instance to additional AWS Regions, you will need to use a customer managed KMS key for encryption at rest. The default AWS owned KMS key type is not supported in a multi-regional IAM Identity Center. For more information, see *Using IAM Identity Center across multiple AWS Regions*.

> **Note**
>
> IAM Identity Center uses envelope encryption in the encryption of your workforce identity data. Your KMS key plays the role of a wrapping key that encrypts the data key that is actually used to encrypt the data.

For more information on AWS KMS, see What is AWS Key Management Service?

IAM Identity Center encryption context

An encryption context is an optional set of non-secret key-value pairs that contain additional contextual information about the data. AWS KMS uses the encryption context as additional

authenticated data to support authenticated encryption. When you include an encryption context in a request to encrypt data, AWS KMS binds the encryption context to the encrypted data. To decrypt data, you include the same encryption context in the request. Refer to the AWS KMS Developer Guide for more information about encryption context.

IAM Identity Center uses encryption context keys from the following: aws:sso:instance-arn, aws:identitystore:identitystore-arn, and tenant-key-id. For example, the following encryption context can appear in AWS KMS API operations invoked by IAM Identity Center API.

```
"encryptionContext": {
    "tenant-key-id": "ssoins-1234567890abcdef",
    "aws:sso:instance-arn": "arn:aws:sso:::instance/ssoins-1234567890abcdef"
}
```

The following encryption context can appear in AWS KMS API operations invoked by Identity Store API.

```
"encryptionContext": {
    "tenant-key-id": "12345678-1234-1234-1234-123456789012",
    "aws:identitystore:identitystore-arn":
 "arn:aws:identitystore::123456789012:identitystore/d-1234567890"
}
```

Using encryption context to control access to your customer managed key

You can use the encryption context in key policies and IAM policies as conditions to control access to your symmetric customer managed key. Some of the key policy templates in the Advanced KMS key policy statements include such conditions to ensure the key is used only with a specific IAM Identity Center instance.

Monitoring your encryption keys for IAM Identity Center

When you use a customer managed KMS key with your IAM Identity Center instance, you can use AWS CloudTrail or Amazon CloudWatch Logs to track requests that IAM Identity Center sends to AWS KMS. The KMS API operations that IAM Identity Center calls are listed in Step 2: Prepare KMS key policy statements. CloudTrail events for these API operations contain the encryption context,

which enables you to monitor AWS KMS API operations called by your IAM Identity Center instance to access data encrypted by your customer managed key.

Example encryption context in a CloudTrail event of an AWS KMS API operation:

```
{
"requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:sso:instance-arn": "arn:aws:sso:::instance/ssoins-xxxxxxxxxxxxxxxx",
            "tenant-key-id": "ssoins-xxxxxxxxxxxxxxxx"
        }
    }
}
```

AWS managed applications' storage, encryption, and deletion of IAM Identity Center identity attributes

Some AWS managed applications that you deploy with AWS IAM Identity Center, such as AWS Systems Manager and Amazon CodeCatalyst, store specific user and group attributes from IAM Identity Center in their own data store. Encryption at rest with a customer managed KMS key in IAM Identity Center does not extend to the IAM Identity Center user and group attributes stored in AWS managed applications. AWS managed applications support different encryption methods for the data they store. Finally, when you delete user and group attributes within IAM Identity Center, these AWS managed applications may continue to store this information past its deletion in IAM Identity Center. Refer to the user guide of your AWS managed applications for encryption and security of data stored within the applications.

Implementing customer managed KMS keys in AWS IAM Identity Center

Customer managed keys are AWS Key Management Service keys that you create, own, and manage. To implement a customer managed KMS key for encryption at rest in AWS IAM Identity Center, follow these steps:

> **Important**
>
> Some AWS managed applications cannot be used with AWS IAM Identity Center configured with a customer managed KMS key. See AWS managed applications that you can use with IAM Identity Center.

1. Step 1: Identify use cases for your organization - To define correct permissions for use of the KMS key you need to identify the relevant use cases across your organization. The KMS key permissions consist of KMS key policy statements and identity-based policies that work together to allow appropriate IAM principals to use the KMS key for their specific use cases.
2. Step 2: Prepare KMS key policy statements - Choose pertinent KMS key policy statement templates based on the use cases identified in Step 1, and fill in required identifiers and IAM principal names. Start with the baseline KMS key policy statements, and if your security policies require it, refine them as described in Advanced KMS key policy statements.
3. Step 3: Create a customer managed KMS key - Create a KMS key in AWS KMS that meets the IAM Identity Center requirements, and add the KMS key policy statements prepared in Step 2 to the KMS key policy.
4. Step 4: Configure IAM policies for cross-account use of the KMS key - Choose pertinent IAM policy statement templates based on the use cases identified in Step 1, and prepare them for use by filling in the key ARN. Then, allow the IAM principals for each specific use case to use the KMS key across accounts by adding the prepared IAM policy statements to the principals' IAM policies.
5. Step 5: Configure the KMS key in IAM Identity Center - Enable the customer managed KMS key in your IAM Identity Center instance to use it for encryption at rest.

Step 1: Identify use cases for your organization

Before creating and configuring your customer managed KMS key, identify your use cases and prepare the required KMS key permissions. Refer to AWS KMS Developer Guide for more information on the KMS key policy.

IAM principals that call the IAM Identity Center service APIs require permissions. For example, a delegated administrator can be authorized to use these APIs through a permission set policy. When IAM Identity Center is configured with a customer managed key, IAM principals must also have permissions to use the KMS API through the IAM Identity Center service APIs. You define these KMS API permissions in two places: the KMS key policy and in the IAM policies associated with the IAM principals.

The KMS key permissions consist of:

1. KMS key policy statements that you specify on the KMS key during its creation in Step 3: Create a customer managed KMS key.

2. IAM policy statements for IAM principals that you specify in Step 4: Configure IAM policies for cross-account use of the KMS key after you create the KMS key.

The following table specifies the relevant use cases and IAM principals that need permissions to use your KMS key.

Use case	IAM principals that need permissions to use the KMS key	Required/ Optional
Use of AWS IAM Identity Center	• Administrators of AWS IAM Identity Center • IAM Identity Center service and the associated Identity Store service	Required
Use of AWS managed applications with IAM Identity Center	• Administrators of AWS managed applications • AWS managed applications • Service roles that AWS managed applications assume to call IAM Identity Center service APIs	Optional
Use of AWS Control Tower on the AWS IAM Identity Center instance it enabled	• AWS Control Tower administrators	Optional
SSO to Amazon EC2 instances with AWS IAM Identity Center	• IAM principals authorized to perform SSO to Amazon EC2 instances	Optional
Any other use case that makes calls to IAM Identity Center service APIs with IAM principals, such as customer managed applications, permission set provisioning workflows, or AWS Lambda functions	• IAM principals used by these workflows to call IAM Identity Center service APIs	Optional

> **Note**
>
> Multiple IAM principals listed in the table require AWS KMS API permissions. However, to protect your user and group data in IAM Identity Center, only IAM Identity Center and Identity Store services directly call the AWS KMS API.

Step 2: Prepare KMS key policy statements

Start with the Baseline KMS key policy and customize it for your organization. If you need more specific policies based on your security requirements, you can modify the policy statements using the examples in Advanced KMS key policy statements. For guidance on this decision, see Considerations for choosing baseline vs. advanced KMS key policy statements.

1. Copy the baseline policy to an editor and insert the required identifiers and IAM principal names in the KMS key policy statements. For help finding the values of the referenced identifiers, see Where to find the required identifiers.
2. If your security requirements call for it, refine the policy statements using the examples in Advanced KMS key policy statements.

> **Important**
>
> Exercise caution when modifying KMS key policies for keys already in use by IAM Identity Center. While IAM Identity Center validates encryption and decryption permissions when you initially configure a KMS key, it cannot verify subsequent policy changes. Inadvertently removing necessary permissions could disrupt your IAM Identity Center's normal operation. For guidance troubleshooting common errors related to customer managed keys in IAM Identity Center, refer to Troubleshoot customer managed keys in AWS IAM Identity Center.

> **Note**
>
> IAM Identity Center and its associated Identity Store require service-level permissions to use your customer managed KMS key. This requirement extends to AWS managed applications that call IAM Identity Center service APIs using service credentials. For other use cases where IAM Identity Center service APIs are called with forward access sessions, only the initiating IAM principal (such as an administrator) needs KMS key permissions.

Notably, end users using the AWS access portal and AWS managed applications don't need direct KMS key permissions, as they are granted through the respective services.

Step 3: Create a customer managed KMS key

You can create a customer managed key using the AWS Management Console or the AWS KMS APIs. While creating the key, add the KMS key policy statements you prepared in Step 2 into the KMS key policy. For detailed instructions, including guidance on the default KMS key policy, see the AWS Key Management Service Developer Guide.

The key must meet the following requirements:

- The KMS key must be in the same AWS Region as the IAM Identity Center instance
- You can choose either a multi-Region or a single-Region key. However, if you plan to use IAM Identity Center in multiple AWS Regions you must create a multi-Region KMS key. You cannot convert a single-Region KMS key to a multi-Region one so we recommend starting with a multi-Region KMS key unless you have specific requirement to use a single-Region KMS key.
- The KMS key must be a symmetric key configured for "encrypt and decrypt" usage
- The KMS key must be in the same AWS Organizations management account as the organization instance of IAM Identity Center

Note

If you plan to replicate this KMS key to Regions where you want to replicate your IAM Identity Center, we recommend you first complete the setup in this section, and then follow the guidance in the section called “Replicate IAM Identity Center to an additional Region”

Step 4: Configure IAM policies for cross-account use of the KMS key

Any IAM principal that uses the IAM Identity Center service APIs from another AWS account, such as IAM Identity Center delegated administrators, also needs an IAM policy statement that allows use of the KMS key through these APIs.

For each use case identified in step 1:

1. Locate the pertinent IAM policy statement templates in Baseline KMS key and IAM policy statements.

2. Copy the templates to an editor and fill in the key ARN, which is now available following the creation of the KMS key in step 3. For help finding the key ARN value, see Where to find the required identifiers.
3. In the AWS Management Console, locate the IAM policy of the IAM principal that is associated with the use case. The location of this policy varies depending on the use case and how access is granted.
   - For access granted directly in IAM, you can locate IAM principals, such as IAM roles in the IAM console.
   - For access granted through IAM Identity Center, you can locate the pertinent permission set in the IAM Identity Center console.
4. Add the use case-specific IAM policy statements to the IAM role and save the change.

> **Note**
>
> The IAM policies described here are identity-based policies. While such policies can be attached to IAM users, groups, and roles, we recommend the use of IAM roles when possible. See the IAM user guide for more information about IAM roles versus IAM users.

Additional configuration in some AWS managed applications

Some AWS managed applications require you to configure a service role to allow the applications to use the IAM Identity Center service APIs. If your organization uses AWS managed applications with IAM Identity Center, complete the following steps for each deployed application:

1. See the application's user guide to confirm whether the permissions have been updated to include KMS key-related permissions for use of the application with IAM Identity Center.
2. If so, update the permissions as instructed in the application's user guide to avoid disruption to the application's operations.

> **Note**
>
> If you're unsure whether an AWS managed application uses these permissions, we recommend that you check the user guides of all deployed AWS managed applications.

You only need to perform this configuration once for each application that requires the configuration.

Step 5: Configure the KMS key in IAM Identity Center

Important

Before proceeding with this step:

- Verify that your AWS managed applications are compatible with customer managed KMS keys. For a list of compatible applications, see AWS managed applications that you can use with IAM Identity Center. If you have incompatible applications, do not proceed.
- Configure the necessary permissions for use of the KMS key. Without proper permissions, this step may fail or disrupt IAM Identity Center administration, the use of AWS managed applications, and other use cases that require KMS key permissions. For more information, see Step 1: Identify use cases for your organization.
- Ensure that permissions for AWS managed applications and customer managed applications that call IAM Identity Center service APIs with IAM roles also allow the use of the KMS key via IAM Identity Center service APIs. Some AWS managed applications require you to configure permissions, such as a service role, for the use of these APIs. Refer to the User Guide of each deployed AWS managed application to confirm whether you need to add specific KMS key permissions.

Specify a KMS key when enabling new organization instance of IAM Identity Center

When enabling a new organization instance of IAM Identity Center, you can specify a customer managed KMS key during setup. This ensures the instance uses your key for encryption at rest from the start. Before you start, refer to Considerations for customer managed KMS keys and advanced KMS key policies.

1. On the **Enable IAM Identity Center** page, expand the **Encryption at rest** section.
2. Choose **Manage Encryption**.
3. Choose **Customer managed key**.
4. For **KMS key**, do one of the following:

   a. Choose **Select from your KMS keys** and select the key you created from the dropdown list.
   b. Choose **Enter KMS key ARN** and enter the full ARN of your key.
5. Choose **Save**.
6. Choose **Enable** to complete the setup.

For more information, see Enable IAM Identity Center.

Change the key configuration for an existing organization instance of IAM Identity Center

You can change your customer managed KMS key to another key or switch to an AWS owned key at any time.

Console

To change your KMS key configuration

1. Open the IAM Identity Center console at https://console.aws.amazon.com/singlesignon/.
2. In the navigation pane, choose **Settings**.
3. Choose the **Additional settings** tab.
4. Choose **Manage encryption**.
5. Choose one of the following:
   a. **Customer managed key** - Select a different customer managed key from the dropdown or enter a new key ARN.
   b. **AWS owned key** - Switch to the default encryption option.
6. Choose **Save**.

AWS CLI

To change an existing organization instance of IAM Identity Center to use KMS customer managed key

```
aws sso-admin update-instance \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
    --encryption-configuration \
```

```
        KeyType=CUSTOMER_MANAGED_KEY,KmsKeyArn=arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

To change an existing organization instance of IAM Identity Center to use AWS owned key

```
aws sso-admin update-instance \
    --instance-arn arn:aws:sso:::instance/ssoins-1234567890abcdef \
    --encryption-configuration KeyType=AWS_OWNED_KMS_KEY
```

Customer managed key considerations

- Updating the KMS key configuring for IAM Identity Center operation has no effect on active user sessions in your IAM Identity Center. You can continue using the AWS access portal, the IAM Identity Center console, and IAM Identity Center service APIs during this process.
- When switching to a new KMS key, IAM Identity Center validates that it can use the key successfully for encryption and decryption. If you made a mistake during the setup of the key policy or IAM policy, the console will show an explanatory error message, and the previous KMS key will remain in use.
- The default annual KMS key rotation will take place automatically. You can refer to the AWS KMS Developer Guide for information on topics such as key rotation, monitoring AWS KMS keys and controlling access to key deletion.

> **Important**
>
> If the customer managed KMS key in use by your IAM Identity Center instance is deleted, disabled, or inaccessible due to an incorrect KMS key policy, your workforce users and IAM Identity Center administrators will not be able to use IAM Identity Center. The loss of access can be temporary (a key policy can be corrected) or permanent (a deleted key cannot be restored) depending on the circumstances. We recommend you restrict access to critical operations, such as deleting or disabling the KMS key. Also, we recommend that your organization set up AWS break-glass access procedures to ensure your privileged users can access AWS in the event that IAM Identity Center is inaccessible.

Where to find the required identifiers

When configuring permissions for your customer managed KMS key, you'll need specific AWS resource identifiers to complete the key policy and IAM policy statement templates. Insert the required identifiers (for example, organization ID) and IAM principal names in the KMS key policy statements.

Below is a guide to locating these identifiers in the AWS Management Console.

IAM Identity Center Amazon Resource Name (ARN) and Identity Store ARN

An IAM Identity Center instance is an AWS resource with its own unique ARN such as arn:aws:sso:::instance/ssoins-1234567890abcdef. The ARN follows the pattern documented in the IAM Identity Center resource types section of the Service Authorization Reference.

Every IAM Identity Center instance has an associated Identity Store that stores the user and group identities. An Identity Store has a unique identifier called Identity Store ID (for example, d-123456789a). The ARN follows the pattern documented in the Identity Store resource types section of the Service Authorization Reference.

You can find both the ARN and the Identity Store ID values on the Settings page of your IAM Identity Center. The Identity store ID is in the Identity source tab.

AWS Organizations ID

If you want to specify an organization ID (for example, o-exampleorg1) in your key policy you can find its value in the Settings page of your IAM Identity Center and Organizations consoles. The ARN follows the pattern documented in the Organizations resource types section of the Service Authorization Reference.

KMS key ARN

You can find the ARN of a KMS key in the AWS KMS console. Choose Customer managed keys on the left, click the key whose ARN you want to look up, and you'll see it in the General configuration section. The ARN follows the pattern documented in the AWS KMS resource types section of the Service Authorization Reference.

See the AWS Key Management Service Developer Guide for more information about Key policies in AWS KMS and troubleshooting AWS KMS permissions. For more information about IAM policies and their JSON representation see the IAM User Guide.

Baseline KMS key policy

The following KMS key policy covers the most common deployment scenario: an IAM Identity Center instance with delegated administrators and AWS managed applications, including AWS Control Tower, SSO to Amazon EC2 instances, and custom workflows. Use this policy as your starting point when creating a customer managed KMS key for IAM Identity Center. If you need more granular access controls, such as restricting the key to a specific IAM Identity Center instance or application, see Advanced KMS key policy statements. Please note if using a multi-Region key, the same policy should be used across all replicas to help ensure consistent authorization.

To use this policy, replace the following placeholder values with your own:

- *111122223333* — The AWS account ID of your IAM Identity Center instance (the AWS Organizations management account).
- *444455556666* — The AWS account ID of your delegated administration account. If you don't use delegated administration, remove this principal.

Since AWS IAM Identity Center requires that the KMS key be in the same AWS Account as the service, the following statements use the `${aws:ResourceOrgID}` and `${aws:ResourceAccount}` variables instead of literal values. You can replace these variables with your AWS Organization ID and AWS Account ID if you prefer to do so.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIdentityCenterAdminAccounts",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root",
          "arn:aws:iam::444455556666:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "AllowIdentityCenterAndIdentityStoreToDescribeKey",
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": [
          "identitystore.amazonaws.com",
          "sso.amazonaws.com"
        ]
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "${aws:ResourceAccount}"
        }
      }
    },
    {
      "Sid": "AllowIdentityCenterAndIdentityStoreToUseKey",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "identitystore.amazonaws.com",
          "sso.amazonaws.com"
        ]
      },
      "Action": [
        "kms:Decrypt",
        "kms:ReEncryptTo",
        "kms:ReEncryptFrom",
        "kms:GenerateDataKeyWithoutPlaintext"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "${aws:ResourceAccount}"
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": [
            "aws:sso:instance-arn",
            "aws:identitystore:identitystore-arn"
          ]
        }
      }
    },
    {
      "Sid": "AllowOrgPrincipalsViaIdentityCenterAndIdentityStore",
```

```
      "Effect": "Allow",
      "Principal": "*",
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "${aws:ResourceOrgID}"
        },
        "StringLike": {
          "kms:ViaService": [
            "sso.*.amazonaws.com",
            "identitystore.*.amazonaws.com"
          ]
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": [
            "aws:sso:instance-arn",
            "aws:identitystore:identitystore-arn"
          ]
        }
      }
    },
    {
      "Sid": "AllowManagedApps",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        },
        "StringEquals": {
          "aws:SourceOrgID": "${aws:ResourceOrgID}"
        },
        "StringLike": {
          "kms:ViaService": [
            "identitystore.*.amazonaws.com",
            "sso.*.amazonaws.com"
          ]
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": [
            "aws:sso:instance-arn",
```

```
            "aws:identitystore:identitystore-arn"
          ]
        }
      }
    }
  ]
}
```

This policy contains five statements. The following table describes what each statement does.

Statement	Purpose
AllowIdentityCenterAdminAccounts	Grants full KMS key permissions to the IAM Identity Center management account and the delegated administration account. This includes key management actions such as modifying the key policy and scheduling key deletion. Administrators in these accounts can manage and use the key if they have the required permissions in their identity-based policies.
AllowIdentityCenterAndIdentityStoreToDescribeKey	Allows the IAM Identity Center and Identity Store service principals to retrieve key metadata. This is required for service operations that validate the key without performing encryption or decryption. The `aws:SourceAccount` condition helps ensure only your IAM Identity Center instance can use your KMS key.
AllowIdentityCenterAndIdentityStoreToUseKey	Allows the IAM Identity Center and Identity Store service principals to use the key for encryption operations such as encrypting, decrypting, and re-encrypting data. The `aws:SourceAccount` condition helps ensure only your IAM Identity Center instance can use your KMS key.
AllowOrgPrincipalsViaIdentityCenterAndIdentityStore	Allows IAM principals in your AWS Organization to decrypt data through the IAM Identity Center and Identity Store services. This covers application administrators who interact with IAM Identity Center-integrated AWS services using Forward Access Sessions (FAS).

Statement	Purpose
AllowManagedApps	Allows AWS managed applications to decrypt data protected by your KMS key through IAM Identity Center and Identity Store.

Use the following IAM policy statement in Step 4: Configure IAM policies for cross-account use of the KMS key to allow delegated administrators to use the KMS key through IAM Identity Center service APIs. Replace the example key ARN with your actual KMS key ARN. The wildcard region in the example accommodates all replicas of a multi-Region KMS key.

For cross-account use cases other than IAM Identity Center administration, such as SSO to Amazon EC2 instances or administration of AWS managed applications, scope `AllowCrossAccountKMSKeyUse` down to `kms:Decrypt` only and remove the `AllowListKMSKeyAliases` statement.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountKMSKeyUse",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:DescribeKey"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/mrk-1234abcd-12ab-34cd-56ef-1234567890ab"
      ]
    },
    {
      "Sid": "AllowListKMSKeyAliases",
      "Effect": "Allow",
      "Action": "kms:ListAliases",
      "Resource": "*"
    }
  ]
}
```

Advanced KMS key policy statements

Use advanced KMS key policy statements to implement more granular access controls for your customer managed KMS key. These policies build on the Baseline KMS key policy by adding encryption context conditions and service-specific restrictions. Before deciding whether to use advanced KMS key policy statements, make sure to review the pertinent considerations.

Using encryption context to restrict access

You can restrict KMS key usage to a specific IAM Identity Center instance by adding encryption context conditions to the `AllowOrgPrincipalsViaIdentityCenterAndIdentityStore` and `AllowManagedApps` statements in your Baseline KMS key policy. Add the following conditions with your specific Identity Center instance ARN and Identity Store ARN. You can also add the same encryption context conditions to the IAM policy configured for cross-account use of the KMS key.

Identity Center

```
"StringEquals": {
    "kms:EncryptionContext:aws:sso:instance-arn":
 "arn:aws:sso:::instance/ssoins-1234567890abcdef"
}
```

Identity Store

```
"StringEquals": {
    "kms:EncryptionContext:aws:identitystore:identitystore-arn":
 "arn:aws:identitystore::111122223333:identitystore/d-1234567890"
}
```

If you need help finding these identifiers, see Where to find the required identifiers .

> **Note**
>
> You can use a customer managed KMS key only with an organization instance of IAM Identity Center. The customer managed key must be located in the AWS organization's management account, which helps ensure the key is used with a single IAM Identity Center instance. However, the encryption context mechanism provides an independent technical safeguard of single-instance usage. You can also use the `aws:SourceArn` condition key in

the KMS key policy statements intended for the Identity Center and Identity Store service principals.

Considerations for implementing encryption context conditions

Before implementing encryption context conditions, review these requirements:

- **DescribeKey action.** The encryption context cannot be applied to the "kms:DescribeKey" action, which can be used by IAM Identity Center administrators. When configuring your KMS key policy, exclude the encryption context for this specific action to ensure proper operations of your IAM Identity Center instance.
- **New instance setup.** If you're enabling a new IAM Identity Center instance with a customer managed KMS key, see Considerations for customer managed KMS keys and advanced KMS key policies.
- **Identity source changes.** When changing your identity source to or from Active Directory, the encryption context requires special attention. See Considerations for changing your identity source.

Policy templates

Choose from these advanced policy templates based on your security requirements. Balance granular access controls with the administrative overhead they introduce.

Topics covered here:

- KMS policy statements for read-only use of a specific IAM Identity Center instance. This section demonstrates the use of the encryption context for read-only access to IAM Identity Center.
- Refined KMS key policy statements for use of AWS managed applications. This section demonstrates how to refine the KMS key policies for AWS managed applications using the encryption context and application information, such as the application service principal, application ARN and AWS account ID.

KMS policy statements for read-only use of a specific IAM Identity Center instance

This policy allows security auditors and other personnel who need only read access to IAM Identity Center to use the KMS key.

To use this policy:

1. Replace the example read-only administrator IAM principals with your actual administrator IAM principals
2. Replace the example IAM Identity Center instance ARN with your actual instance ARN
3. Replace the example Identity Store ARN with your actual Identity Store ARN
4. If using delegated administration, see Step 4: Configure IAM policies for cross-account use of the KMS key

If you need help finding the values of these identifiers, see Where to find the required identifiers .

Once you have updated the template with your values, return to Step 2: Prepare KMS key policy statements to prepare additional KMS key policy statements, as needed.

The kms:Decrypt action alone does not restrict access to read-only operations. The IAM policy must enforce read-only access on IAM Identity Center service APIs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowReadOnlyAccessToIdentityCenterAPI",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyAdminRole"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "sso.*.amazonaws.com",
          "kms:EncryptionContext:aws:sso:instance-arn":
 "arn:aws:sso:::instance/ssoins-1234567890abcdef"
        }
      }
    },
    {
      "Sid": "AllowReadOnlyAccessToIdentityStoreAPI",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/MyAdminRole"
```

```
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "identitystore.*.amazonaws.com",
          "kms:EncryptionContext:aws:identitystore:identitystore-arn":
 "arn:aws:identitystore::111122223333:identitystore/d-1234567890"
        }
      }
    }
  ]
}
```

Refined KMS key policy statements for use of AWS managed applications

These policy templates provide more granular control over which AWS managed applications can use your KMS key.

> **Note**
>
> Some AWS managed applications cannot be used with IAM Identity Center configured with a customer managed KMS key. See AWS managed applications that you can use with IAM Identity Center.

The Baseline KMS key policy allows any AWS managed application from any account in the same AWS organization to use the KMS key. Use these refined policies to restrict access by:

- Application service principal
- Application instance ARNs
- AWS account IDs
- Encryption context for specific IAM Identity Center instances

> **Note**
>
> A service principal is a unique identifier for an AWS service, typically formatted as servicename.amazonaws.com (for example, elasticmapreduce.amazonaws.com for Amazon EMR).

Restrict by account

This KMS key policy statement template allows an AWS managed application in specific AWS accounts to use the KMS key using a specific IAM Identity Center instance.

To use this policy:

1. Replace the example service principal with your actual application service principal
2. Replace the example account IDs with the actual account IDs where your AWS managed applications are deployed
3. Replace the example Identity Store ARN with your actual Identity Store ARN
4. Replace the example IAM Identity Center instance ARN with your actual instance ARN

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowServiceInSpecificAccountsToUseTheKMSKeyViaIdentityCenter",
      "Effect": "Allow",
      "Principal": {
        "Service": "myapp.amazonaws.com"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "111122223333",
            "444455556666"
          ]
        },
        "StringLike": {
          "kms:ViaService": "sso.*.amazonaws.com",
```

```
          "kms:EncryptionContext:aws:sso:instance-arn": "arn:aws:sso:::instance/
ssoins-1234567890abcdef"
        },
        "Bool": {
          "aws:PprincipalIsAWSService": "true"
        }
      }
    },
    {
      "Sid": "AllowServiceInSpecificAccountsToUseTheKMSKeyViaIdentityStore",
      "Effect": "Allow",
      "Principal": {
        "Service": "myapp.amazonaws.com"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "111122223333",
            "444455556666"
          ]
        },
        "StringLike": {
          "kms:ViaService": "identitystore.*.amazonaws.com",
          "kms:EncryptionContext:aws:identitystore:identitystore-arn":
 "arn:aws:identitystore::111122223333:identitystore/d-1234567890"
        },
        "Bool": {
          "aws:PprincipalIsAWSService": "true"
        }
      }
    }
  ]
}
```

Restrict by application instance

This KMS key policy statement template allows a specific AWS managed application instance to use the KMS key using a specific IAM Identity Center instance.

To use this policy:

1. Replace the example service principal with your actual application service principal

2. Replace the example application ARN with your actual application instance ARN
3. Replace the example Identity Store ARN with your actual Identity Store ARN
4. Replace the example IAM Identity Center instance ARN with your actual instance ARN

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSpecificAppInstanceToUseTheKMSKeyViaIdentityCenter",
      "Effect": "Allow",
      "Principal": {
        "Service": "myapp.amazonaws.com"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceARN": "arn:aws:myapp:us-east-1:111122223333:application/my-
application"
        },
        "StringLike": {
          "kms:ViaService": "sso.*.amazonaws.com",
          "kms:EncryptionContext:aws:sso:instance-arn": "arn:aws:sso:::instance/
ssoins-1234567890abcdef"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        }
      }
    },
    {
      "Sid": "AllowSpecificAppInstanceToUseTheKMSKeyViaIdentityStore",
      "Effect": "Allow",
      "Principal": {
        "Service": "myapp.amazonaws.com"
      },
      "Action": "kms:Decrypt",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceARN": "arn:aws:myapp:us-east-1:111122223333:application/my-
application"
```

```
        },
        "StringLike": {
          "kms:ViaService": "identitystore.*.amazonaws.com",
          "kms:EncryptionContext:aws:identitystore:identitystore-arn":
 "arn:aws:identitystore::111122223333:identitystore/d-1234567890"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        }
      }
    }
  ]
}
```

Considerations for customer managed KMS keys and advanced KMS key policies

When implementing customer managed KMS keys with IAM Identity Center, consider these factors that affect setup, security, and ongoing maintenance of your encryption configuration.

Considerations for choosing baseline vs. advanced KMS key policy statements

When deciding whether to make the KMS key permissions more specific using Advanced KMS key policy statements, consider the management overhead and the security needs of your organization. More specific policy statements provide finer-grained control over who can use the key and for what purposes; however, they require ongoing maintenance as your IAM Identity Center configuration evolves. For example, if you restrict the use of the KMS key to specific AWS managed application deployments, you'll need to update the key policy whenever your organization wants to deploy or undeploy an application. Less restrictive policies reduce administrative burden but may grant broader permissions than necessary for your security requirements.

Considerations for enabling a new IAM Identity Center instance with a customer managed KMS key

The considerations here apply if you're using the encryption context as described in Advanced KMS key policy statements to restrict use of the KMS key to a specific IAM Identity Center instance.

When enabling a new IAM Identity Center instance with a customer managed KMS key, the IAM Identity Center and Identity Store ARNs are not available until after setup. You have the following options:

- Use generic ARN patterns temporarily, and then replace with full ARNs after the instance is enabled. Remember to switch between StringEquals and StringLike operators as needed.
  - For IAM Identity Center SPN: "arn:${Partition}:sso:::instance/*".
  - For Identity Store SPN: "arn:${Partition}:identitystore::${Account}:identitystore/*".
- Use "purpose:KEY_CONFIGURATION" in the ARN temporarily. This works only for instance enablement and must be replaced with the actual ARN for your IAM Identity Center instance to function normally. The advantage of this approach is that you cannot forget to replace this after the instance is enabled.
  - For IAM Identity Center SPN, use: "arn:${Partition}:sso:::instance/purpose:KEY_CONFIGURATION"
  - For Identity Store SPN, use: "arn:${Partition}:identitystore::${Account}:identitystore/purpose:KEY_CONFIGURATION"

> **Important**
>
> Don't apply this configuration to a KMS key already in use in an existing IAM Identity Center instance, as it may disrupt its normal operations.

- Omit the encryption context condition from the KMS key policy until after the instance is enabled.

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 cannot 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.

You can apply tags to permission sets only. You cannot 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 is 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,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
```

Manage tags using the IAM Identity Center API

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 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, 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*.

Considerations for AWS Management Console Private Access

If your organization uses the AWS Management Console Private Access feature, you should consider how your users will sign-in to IAM Identity Center.

A VPC endpoint policy restricts sign-in to the management console, which prevents your users from signing in to AWS accounts they are not authorized to access. For more information, see AWS Management Console Private Access in the *AWS Management Console Getting Started Guide*.

VPC endpoints block sign-in to the IAM Identity Center

It's important to note that using VPC endpoints will block sign-in to the IAM Identity Center. This happens when a user is already logged into the management console through the VPC endpoint. To ensure your users can continue to sign-in to IAM Identity Center, they must use the public endpoint for AWS sign-in, rather than the VPC endpoint.

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 to administer IAM Identity Center 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	3500	Yes
Number of provisioned permission sets allowed per AWS account	500	Yes
Number of inline policies per permission set	1	No
Number of AWS managed and customer managed policies per permission set	25[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 (permissi on sets) in the AWS account that can be updated at a time	1	No

[1]AWS 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	200000	Yes
Number of groups supported in IAM Identity Center	100000	Yes
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 limit of 20 transactions per second

Resource	Default quota
	(TPS). For read APIs, you can open a support case to request a limit increase. The CreateAccountAssignment write API has a limit of 15 outstanding asynchronous calls. This limit cannot be increased.
Identity Store APIs	Identity Store APIs have a throttle limit of 20 transactions per second (TPS) per API. This limit applies per Identity Store instance. You can open a support case to request a limit increase.
SCIM APIs	SCIM APIs have per-API throttle limits of 25 transactions per second (TPS) for write APIs and 40 TPS for read APIs. These limits apply per Identity Store instance. You can open a support case to request a limit increase. Each individual membership operation in `CreateGroup` or `PatchGroup` calls counts as a transaction toward a separate membership operation throttle limit per Identity Store.

If your IAM Identity Center instance is enabled in multiple AWS Regions, the throttle limits apply equally to each enabled Region. For example, you would have the 20 TPS throttle limit on the Identity Store APIs in each enabled Region. For more information about which API operations are available in additional Regions, see the corresponding table.

OIDC service request quotas

Resource	Default value (requests per second)	Can be increased
Request rate from a remote address to register a public OAuth client Applies to: RegisterClient	20	Yes
Request rate from a public client registered with the OIDC service Applies to: CreateToken, StartDeviceAuthorization	80	Yes
Request rate from all public clients registered with the same IAM Identity Center instance Applies to: CreateToken	250	Yes
Request rate from an IAM Identity Center applicati on registered with the IAM Identity Center instance with Token Exchange and Refresh Token grants. Applies to: CreateTok enWithIAM	80	Yes
Token generation rate from all IAM Identity Center applications registered with the same IAM Identity Center	10	Contact AWS Support

Resource	Default value (requests per second)	Can be increased
instance with JWT Bearer grant Applies to: CreateTokenWithIAM		

If your IAM Identity Center instance is enabled in multiple AWS Regions, the request rates above apply equally to each enabled Region. For example, if your allowed request rate to register a public OAuth client from a remote address is 20 requests per second, this throughput is available in each enabled Region. For more information about which API operations are available in additional Regions, see the corresponding table.

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
Total number of groups that can be assigned to a permission set per AWS account, or to an application	100	No

Resource	Default quota	Can be increased
Total number of AWS Regions enabled for a single IAM Identity Center instance	3	Yes

* For example, you might configure 2750 accounts and 250 applications, resulting in a total of 3000 accounts and applications.

** The`ProvisionPermissionSet` API operation can provision a permission set using the option `ALL_PROVISIONED_ACCOUNTS` to, at most, 3500 AWS accounts. If you need to provision a permission set to more than 3500 AWS accounts, you can use the `ProvisionPermissionSet` API operation with the `AWS_ACCOUNT` option, which provisions the permission set in a single AWS account. You can make up to three concurrent calls to `ProvisionPermissionSet`.

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 are 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 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 your Identity Provider (IdP) is configured to provision users into IAM Identity Center using SCIM synchronization, you may encounter synchronization failures during the user provisioning process. This may indicate that the user configuration in your IdP is not compatible with the IAM Identity Center requirements. When this happens, the IAM Identity Center SCIM APIs will return error messages that provide insights into the root cause of the issue. You can locate these error messages in your IdP's logs or UI. Alternatively, you may find more detailed information about the provisioning failures in the AWS CloudTrail logs.

For more information on the IAM Identity Center SCIM implementations, including the specifications of required, optional, and unsupported parameters and operations for user objects, see IAM Identity Center SCIM Implementation Developer Guide in the *SCIM Developer Guide*

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.

Error Message: "2 validation errors detected: Value at *'name.givenName'* failed to satisfy constraint: Member must satisfy regular expression pattern: [\\p{L}\\p{M}\\p{S}\\p{N}\\p{P}\\t\\n\\r]+; Value at *'name.givenName'* failed to satisfy constraint: Member must have length greater than or equal to 1"

- **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.

   Error Message: "List attribute *emails* exceeds allowed limit of 1"

   - **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.

Duplicate user or group error when provisioning users or groups with an external identity provider

If you experience IAM Identity Center synchronization issues when provisioning users or groups in an external identity provider (IdP), it could be due to your external IdP users or groups not having unique attribute values. You may receive the following error messages in your external IdP:

Refused to create a new, duplicate resource

You can experience this problem in the following scenarios:

- **Scenario 1**
  - You're using customized non-unique attributes in your external IdP for attributes that must be unique in IAM Identity Center. Existing IAM Identity Center users or groups fail to synchronize to your IdP.
- **Scenario 2**
  - You attempt to create users that have duplicate attributes for attributes that must be unique in IAM Identity Center.
    - For example, you create or have an existing IAM Identity Center user with the following attributes:
      - **Username:** Jane Doe
      - **Primary Email Address:** `jane_doe@example.com`
    - Then you attempt to create another user in your external IdP with the following attributes:
      - **Username:** Richard Doe
      - **Primary Email Address:** `jane_doe@example.com`
        - The external IdP attempts to synchronize and create the user in IAM Identity Center. However, these actions fail as both users have duplicate values for a primary email address which must be unique.

The username, primary email address, and externalID must be unique in order for your external IdP users to successfully synchronize to IAM Identity Center. Similarly, the group name must be unique for your external IdP groups to successfully synchronize to IAM Identity Center.

The solution is to review your identity source's attributes and ensure they are unique.

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 MFA for Identity Center directory 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 cannot 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 is 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 are 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:names: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 cannot 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 is 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.

Group members from trusted domains do not sync to IAM Identity Center

If a security group syncs successfully to IAM Identity Center, but its members from a trusted on-premises domain do not appear in the IAM Identity Center, this can be caused by the group resides

in AWS Managed Microsoft AD and contains Foreign Security Principal (FSPs) representing users from a trusted domain.

Try the following steps to troubleshoot:

- Sync groups directly from the trusted domain: Instead of syncing a group from AWS Managed Microsoft AD that contains cross-domain members, create and sync the group directly from the trusted on-premises domain. This approach works because IAM Identity Center can access actual user objects in the source domain.
- Verify service account permissions: Ensure the IAM Identity Center service account has the required permissions on user objects in the trusted domain: ReadProperties and ListContents permission.

Troubleshoot customer managed keys in AWS IAM Identity Center

This topic describes common customer managed key related errors you might encounter when using AWS IAM Identity Center and provides troubleshooting steps to resolve them.

Access Denied: KMS Decrypt Permission Issue

Error: "User xxxxxxx is not authorized to perform: kms:Decrypt on the resource associated with this ciphertext because no identity-based policy allows the kms:Decrypt action"

The user or IAM principal lacks the required `kms:Decrypt` permission in either their IAM policy or KMS key policy.

Troubleshooting with AWS CloudTrail:

1. Look for `kms.amazonaws.com` events in CloudTrail
2. Search for event name `Decrypt`
3. Review the `errorCode` and `errorMessage` fields
4. Check `userIdentity` to confirm which principal attempted the operation

To resolve this issue, grant the user or IAM principal `kms:Decrypt` access permissions in their IAM policy and KMS key policy. For more information, see Implementing customer managed KMS keys in AWS IAM Identity Center.

AWS managed application login failures with a customer managed KMS key enabled in IAM Identity Center

If no Identity Center users can log into AWS managed applications and you have a customer managed KMS key enabled in your IAM Identity Center instance, verify that the KMS key policy grants the AWS managed applications permissions to use the customer managed KMS key. For more information, see Baseline KMS key policy.

AWS managed application installation and/or user assignment failures with a customer managed KMS key enabled in IAM Identity Center

Error: "User xxxxxxx is not authorized to perform: kms:Decrypt on the resource associated with this ciphertext because no identity-based policy allows the kms:Decrypt action"

The user or IAM principal lacks the required `kms:Decrypt` permission in either their IAM policy or KMS key policy.

Troubleshooting with CloudTrail:

1. Search for event name `Decrypt`
2. Review the `errorCode` and `errorMessage` fields
3. Check `userIdentity` to confirm which principal attempted the operation

To resolve this issue, grant the user or IAM principal `kms:Decrypt` access permissions in their IAM policy and KMS key policy. For more information, see Implementing customer managed KMS keys in AWS IAM Identity Center.

KMS Permissions Issue: Configuring Customer Managed Key with AWS IAM Identity Center

The user or IAM principal lacks one or more required KMS permissions (`kms:Decrypt`, `kms:Encrypt`, `kms:GenerateDataKey`, `kms:DescribeKey`) when enabling customer managed key.

Troubleshooting with CloudTrail:

1. Search for `Decrypt`, `Encrypt`, `GenerateDataKey`, or `DescribeKey` events

2. Review the `errorCode` and `errorMessage` fields
3. Check `userIdentity` to confirm which principal attempted the operation

To resolve this issue, grant all required KMS permissions to the user or IAM principal in their identity-based policy or KMS key policy. For more information, see Implementing customer managed KMS keys in AWS IAM Identity Center.

AWS access portal login failures with a customer managed KMS key enabled in IAM Identity Center

Error: "ERROR Code: 0001 - IdentityCenter service access is blocked. Reach out to your IdentityCenter admin for further steps."

If users cannot log in to the AWS access portal and you have a customer managed KMS key enabled in your IAM Identity Center instance, verify that the KMS key policy grants the necessary permissions to Identity Center and Identity Store. For more information, see Baseline KMS key policy.

Troubleshoot multi-Region setup in AWS IAM Identity Center

This topic describes common multi-Region setup related errors you might encounter when using AWS IAM Identity Center and provides troubleshooting steps to resolve them.

The Region I want to replicate my IAM Identity Center instance to is not available in the IAM Identity Center console

You must first create a replica key for your customer managed KMS key in the Region you want to replicate your IAM Identity Center instance to. Once the replica key is created, you will see the Region in the list of Regions available for replications. For more information, see the section called “Step 1: Create a replica key”.

AWS managed application sign-in failures in an additional Region

If no IAM Identity Center users can sign into AWS managed applications in an additional Region after you added the Region in IAM Identity Center, confirm that you configured the additional Region's Assertion Consumer Service (ACS) URL in the external identity provider as described in the

section called “Step 3: Update external IdP setup”. Also, confirm your users have connectivity to the Region.

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:** October 21, 2025

Change	Description	Date
Simplified baseline KMS key policy	Combined the baseline KMS key policy statements into a single simplified policy, which streamlines the configuration of the customer managed KMS key for encryption at rest.	April 30, 2026
Multi-Region support	Added multi-Region support, including replication to additional Regions, AWS account access and use of AWS managed applications in additional Regions.	February 2, 2026
Updated AWS managed policy	Updated the `AWSIdentityCenterExternalManagementPolicy` AWS managed policy to change the ARN for provisioning tenant.	December 5, 2025
New AWS managed policy topic	Added details for the `AWSIdentityCenterExternalManagementPolicy` AWS managed policy.	November 21, 2025
Auto-provisioning audit and reconciliation guidance	Added guidance for auditing and reconciling SCIM auto-	October 17, 2025

	provisioned users, groups, and group memberships using Identity Store APIs and AWS CLI commands.	
OIDC service request quotas	Added OIDC service request quotas, including rate limits for OAuth client registration, token creation, and other OIDC operations.	October 13, 2025
Updates for AWS managed policy	IAM Identity Center has updated the managed policies `AWSSSOMasterAccountAdministrator`, `AWSSSOMemberAccountAdministrator`, `AWSSSOReadOnly`, `AWSSSODirectoryAdministrator`, and `AWSSSODirectoryReadOnly` to include AWS KMS permissions required for IAM Identity Center instances that use customer managed keys for encryption.	September 17, 2025
Customer managed keys for encryption at rest	Added support for customer managed KMS keys to encrypt workforce identity data at rest.	September 17, 2025
Example resource-based policy to allow token creation	Added example resource-based policy that allows token creation for authorized client appplications.	September 16, 2025

Support for user background sessions	Added content for user background sessions.	August 11, 2025
Identity-enhanced console sessions	Updated terminology for identity-enhanced console sessions (previously known as identity-aware sessions).	May 12, 2025
Getting started reorganization	Reorganized getting started content for improved clarity and user experience.	May 6, 2025
Deprecate IAM Identity Center AD Sync	You can no longer provision Active Directory users with IAM Identity Center AD Sync. Instead, you can use IAM Identity Center configurable AD Sync.	April 17, 2025
Updated content for Authenticated session	Update to IAM Identity Center session durations when user session is deleted.	April 2, 2025
Updates for AWS managed policy	Updated permissions for the `AWSSSOServiceRolePolicy` AWS managed policy.	February 11, 2025
IAM Identity Center enablement workflow improved	Updated workflow for enabling organization instances and account instances of IAM Identity Center.	February 11, 2025
Updates for IAM Identity Center enablement	Updated contents and procedures for enabling organization instances and account instances of IAM Identity Center.	October 10, 2024

Updates for AWS managed policy	Updated permissions for the `AWSIAMIdentityCenterAllowListForIdentityContext` AWS managed policy.	October 2, 2024
Updates for AWS managed policy	Updated permissions for the `AWSSSOMasterAccountAdministrator` AWS managed policy.	September 26, 2024
Updates for AWS managed policy	Updated permissions for the `AWSIAMIdentityCenterAllowListForIdentityContext` AWS managed policy.	September 4, 2024
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
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 `AWSSSOReadOnly` 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 `AWSSSOServiceRolePolicy` 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 `AWSSSOMasterAccountAdministrator`, `AWSSSOMemberAccountAdministrator`, and `AWSSSOReadOnly` 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 `AWSSSOMasterAccountAdministrator` 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 verificat ion 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 applicati ons	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, DruvaInSync, 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*.