# Table of Contents

What Is AWS Single Sign-On? .......................................................................................................................... 1
AWS SSO Features ........................................................................................................................................... 1
Getting Started .................................................................................................................................................. 3
AWS SSO Prerequisites ................................................................................................................................. 3
Step 1: Enable AWS SSO ............................................................................................................................... 3
Step 2: Choose Your Identity Source ........................................................................................................... 4
Step 3: Set Up SSO to Your AWS Accounts ................................................................................................. 4
Step 4: Set Up SSO to Your Applications .................................................................................................... 4
Key AWS SSO Concepts .................................................................................................................................. 6
Users, Groups, and Provisioning .................................................................................................................... 6
  User name and email address uniqueness ................................................................................................. 6
  Groups .......................................................................................................................................................... 6
  User and group provisioning ....................................................................................................................... 6
AWS SSO-Integrated Application Enablement ............................................................................................... 7
  Considerations for Sharing Identity Information in AWS Accounts ....................................................... 7
  Enable AWS SSO-Integrated Applications in AWS Accounts ................................................................. 7
SAML Federation ............................................................................................................................................... 8
User Authentications ....................................................................................................................................... 8
Permission Sets ................................................................................................................................................ 8
Manage Your Identity Source ........................................................................................................................ 9
  Considerations for Changing Your Identity Source ............................................................................... 9
    Changing Between AWS SSO and Active Directory ....................................................................... 9
    Changing Between AWS SSO and Azure Active Directory ............................................................... 9
  Change Your Identity Source .................................................................................................................. 10
Manage Identities in AWS SSO ..................................................................................................................... 10
  Provisioning When Users are in AWS SSO ............................................................................................. 10
  Add Users .................................................................................................................................................. 11
  Add Groups ............................................................................................................................................... 11
  Add Users to Groups ................................................................................................................................. 12
  Edit User Properties ................................................................................................................................. 12
  Disable a User ........................................................................................................................................... 12
  Reset a User Password ............................................................................................................................. 13
  Password Requirements ........................................................................................................................... 13
  Supported User and Group Attributes ..................................................................................................... 13
Connect to Your Microsoft AD Directory .................................................................................................... 14
  Provisioning When Users Come from Active Directory ....................................................................... 14
  Connect AWS SSO to an AWS Managed Microsoft AD Directory ......................................................... 15
  Connect AWS SSO to an On-Premises Active Directory .................................................................... 15
  Attribute Mappings .................................................................................................................................. 16
Connect to Your External Identity Provider ................................................................................................ 18
  Provisioning When Users Come from an External Identity Provider .................................................. 19
  How to Connect to an External Identity Provider ............................................................................... 19
  Automatic Provisioning ............................................................................................................................ 20
  Manual Provisioning ................................................................................................................................. 22
Manage SSO to Your AWS Accounts .......................................................................................................... 23
  Single Sign-On Access .............................................................................................................................. 23
    Assign User Access ............................................................................................................................... 24
    Remove User Access ............................................................................................................................. 24
    Delegate Who Can Assign SSO Access to Users in the Master Account ........................................... 25
  Permission Sets ....................................................................................................................................... 25
    Create Permission Set ........................................................................................................................... 25
    Configure Permission Set Properties ................................................................................................. 26
    Delete Permission Sets .......................................................................................................................... 28
What Is AWS Single Sign-On?

AWS Single Sign-On is a cloud-based single sign-on (SSO) service that makes it easy to centrally manage SSO access to all of your AWS accounts and cloud applications. Specifically, it helps you manage SSO access and user permissions across all your AWS accounts in AWS Organizations. AWS SSO also helps you manage access and permissions to commonly used third-party software as a service (SaaS) applications, AWS SSO-integrated applications as well as custom applications that support Security Assertion Markup Language (SAML) 2.0. AWS SSO includes a user portal where your end-users can find and access all their assigned AWS accounts, cloud applications, and custom applications in one place.

AWS SSO Features

AWS SSO provides the following features:

Integration with AWS Organizations

AWS SSO is integrated deeply with AWS Organizations and AWS API operations, unlike other cloud native SSO solutions. AWS SSO natively integrates with AWS Organizations and enumerates all your AWS accounts. If you have organized your accounts under organizational units (OUs) you will see them displayed that way within the AWS SSO console. That way you can quickly discover your AWS accounts, deploy common sets of permissions, and manage access from a central location.

SSO access to your AWS accounts and cloud applications

AWS SSO makes it simple for you to manage SSO across all your AWS accounts, cloud applications, AWS SSO-integrated applications, and custom SAML 2.0–based applications, without custom scripts or third-party SSO solutions. Use the AWS SSO console to quickly assign which users should have one-click access to only the applications that you've authorized for their personalized end-user portal.

Create and manage users and groups in AWS SSO

When you enable the service for the first time, we create a default store for you in AWS SSO. You can use this store to manage your users and groups directly in the console. Or, if you prefer, you can connect to an existing AWS Managed Microsoft AD directory and manage your users with standard Active Directory management tools provided in Windows Server. You can also provision users and groups from an external identity provider into AWS SSO and manage access permissions in the AWS SSO console. If you choose to manage your users in AWS SSO, you can quickly create users and then easily organize them into groups, all within the console.

Leverage your existing corporate identities

AWS SSO is integrated with Microsoft AD through the AWS Directory Service. That means your employees can sign in to your AWS SSO user portal using their corporate Active Directory credentials. To grant Active Directory users access to accounts and applications, you simply add them to the appropriate Active Directory groups. For example, you can grant the DevOps group SSO access to your production AWS accounts. Users added to the DevOps group are then granted SSO access to these AWS accounts automatically. This automation makes it easy to onboard new users and give existing users access to new accounts and applications quickly.

Compatible with commonly used cloud applications

AWS SSO supports commonly used cloud applications such as Salesforce, Box, and Office 365. This cuts the time needed to set up these applications for SSO by providing application integration
instructions. These instructions act as guard rails to help administrators set up and troubleshoot these SSO configurations. This eliminates the need for administrators to learn the configuration nuances of each cloud application.

**Easy to set up and monitor usage**

With AWS SSO, you can enable a highly available SSO service with just a few clicks. There is no additional infrastructure to deploy or AWS account to set up. AWS SSO is a highly available and a completely secure infrastructure that scales to your needs and does not require software or hardware to manage. AWS SSO records all sign-in activity in AWS CloudTrail, giving you the visibility to monitor and audit SSO activity in one place.
Getting Started

In this getting started exercise, you enable AWS Single Sign-On, connect your directory, set up SSO to your AWS accounts, and finally set up SSO to your cloud applications. Although not required, we recommend that you review Understanding Key AWS Single Sign-On Concepts (p. 6) before you begin using the console so that you are familiar with the core features and terminology.

Topics
- AWS SSO Prerequisites (p. 3)
- Enable AWS SSO (p. 3)
- Choose Your Identity Source (p. 4)
- Set Up SSO to Your AWS Accounts (p. 4)
- Set Up SSO to Your Applications (p. 4)

AWS SSO Prerequisites

Before you can set up AWS SSO, you must:

- Have first set up the AWS Organizations service and have All features set to enabled. For more information about this setting, see Enabling All Features in Your Organization in the AWS Organizations User Guide.
- Sign in with the AWS Organizations master account credentials before you begin setting up AWS SSO. These credentials are required to enable AWS SSO. For more information, see Creating and Managing an AWS Organization in the AWS Organizations User Guide. You cannot set up AWS SSO while signed in with credentials from an Organization's member account.
- Have chosen an identity source to determine which pool of users has SSO access to the user portal. If you choose to use the default AWS SSO identity source for your user store, no prerequisite tasks are required. The AWS SSO store is created by default once you enable AWS SSO and is immediately ready for use. There is no cost for using this store. Alternatively, you can choose to Connect to Your External Identity Provider (p. 18) using Azure Active Directory. If you choose to connect to an existing Active Directory for your user store, you must have the following:
  - An existing AD Connector or AWS Managed Microsoft AD directory set up in AWS Directory Service, and it must reside within your organization's master account. You can connect only one AWS Managed Microsoft AD directory at a time. However, you can change it to a different AWS Managed Microsoft AD directory or change it back to an AWS SSO store at any time. For more information, see Create a AWS Managed Microsoft AD Directory in the AWS Directory Service Administration Guide.
  - You must set up AWS SSO in the Region where your AWS Managed Microsoft AD directory is set up. AWS SSO stores the assignment data in the same Region as the directory. To administer AWS SSO, you should switch to the Region where you have setup AWS SSO. Also, note that AWS SSO’s user portal uses the same access URL as your connected directory.

Enable AWS SSO

When you open the AWS SSO console for the first time, you are prompted to enable AWS SSO before you can start managing it. If you have already chosen this option, you can skip this step. If not, use the procedure below to enable it now. Once enabled, AWS SSO is granted the necessary permissions to create IAM service-linked roles in any of the AWS accounts within your AWS organization. No service-
linked roles are created at this time. AWS SSO creates these roles later during the process of setting up SSO access to your AWS accounts (see Set Up SSO to Your AWS Accounts (p. 4)).

To enable AWS SSO

1. Sign in to the AWS Management Console with your AWS Organizations master account credentials.
2. Open the AWS SSO console.
3. Choose Enable AWS SSO.
4. If you have not yet set up AWS Organizations, you will be prompted to create an organization. Choose Create AWS organization to complete this process.

Choose Your Identity Source

Choosing an identity source determines where AWS SSO looks for users and groups that need SSO access. By default, you get an AWS SSO store for quick and easy user management. Optionally, you can also connect an external identity provider or connect an AWS Managed Microsoft AD directory with your on-premises Active Directory.

AWS SSO provides users in this identity source with a personalized user portal from which they can easily launch multiple AWS accounts or cloud applications. Users sign in to the portal using their corporate credentials or with credentials they set up in AWS SSO. Once they sign in, they have one-click access to all applications and AWS accounts that you have previously authorized.

Depending on which identity source type you are trying to set up, review the topics below for guidance:

- Manage Identities in AWS SSO (p. 10)
- Connect to Your Microsoft AD Directory (p. 14)
- Connect to Your External Identity Provider (p. 18)

For more information about supported identity source types, see Manage Your Identity Source (p. 9).

Set Up SSO to Your AWS Accounts

In this step, you can grant users in your directory with SSO access to one or more AWS consoles for specific AWS accounts in your AWS organization. Afterward, users see only the AWS account icon (for example, Development) that they’ve been assigned from within their user portal. When they click the icon, they can then choose which IAM role they want to use when signing in to the AWS Management Console for that AWS account.

To get started assigning SSO access to your AWS accounts, see Assign User Access (p. 24).

Set Up SSO to Your Applications

With AWS SSO, you can use AWS applications that are integrated with AWS SSO, cloud-applications for which AWS provides preintegration, and custom SAML 2.0 applications. Depending on which application type you are trying to set up, review the topics below:

- Add and Configure an AWS SSO-Integrated Application (p. 31)
- Add and Configure a Cloud Application (p. 34)
- Add and Configure a Custom SAML 2.0 Application (p. 35)
For more information about supported application types, see Manage SSO to Your Applications (p. 30).

After you follow the guidance in the topic, you will have successfully configured AWS SSO and set up a trust with your service provider. Your users can now access these applications from within their user portal based on the permissions you assigned.
Understanding Key AWS Single Sign-On Concepts

You'll get more out of AWS Single Sign-On if you become familiar with key concepts relating to SAML federation, user authentication, and IAM permissions.

Topics
- Users, Groups, and Provisioning (p. 6)
- AWS SSO-Integrated Application Enablement (p. 7)
- SAML Federation (p. 8)
- User Authentications (p. 8)
- Permission Sets (p. 8)

Users, Groups, and Provisioning

AWS SSO manages access to all your AWS Organizations accounts, AWS SSO-integrated applications, and other business applications that support the Security Assertion Markup Language (SAML) 2.0 standard.

User name and email address uniqueness

When working in AWS SSO, users must be uniquely identifiable. AWS SSO implements a user name that is the primary identifier for your users. Although most people set the user name equal to a user's email address, AWS SSO and the SAML standard do not require this. However, a large percentage of SAML-based applications use an email address as the unique identifier for users. They obtain this from assertions that a SAML identity provider sends during authentication. Such applications depend upon the uniqueness of email addresses for each user. As such, AWS SSO allows you to specify something other than an email address for user sign-in. AWS SSO requires that all user names and email addresses for your users are non-NULL and unique.

Groups

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

User and group provisioning

You can create users and groups directly in AWS SSO, or work with users and groups you have in Active Directory or an external identity provider. In order for AWS SSO to assign users and groups for permissions in an AWS SSO account, AWS SSO must first be aware of the users and groups. Similarly, AWS SSO-integrated applications can work with users and groups for which AWS SSO is aware. Provisioning is the process of making user and group information available for use by AWS SSO and AWS SSO-integrated applications.
Provisioning in AWS SSO varies based on the identity source you use. For more information, see Manage Your Identity Source (p. 9).

## AWS SSO-Integrated Application Enablement

AWS SSO provides support for integration by other AWS applications and services. These applications can use AWS SSO to perform authentication and can access information about users and groups. For example, a user might sign into an application that generates performance dashboards for resources that the user controls. The user might then share the dashboard by looking up a group in AWS SSO.

To enable this capability, AWS SSO contains an identity store that contains user and group attributes, excluding sign-in credentials. These attributes have several things in common:

- Synchronized (provisioned) automatically via just-in-time (JIT) provisioning when you use Microsoft Active Directory as your identity source
- Provisioned continuously and automatically when you configure a System for Cross-domain Identity Management (SCIM) 2.0 connection to AWS SSO
- Provisioned manually from the AWS SSO console when you use an external identity provider without SCIM 2.0 automatic provisioning
- Created in AWS SSO when you use AWS SSO as your identity source

Regardless of what you use as your identity source, AWS SSO has the ability to share the user and group information with AWS SSO-integrated applications. This capability makes it possible to connect an identity source to AWS SSO once and then share identity information with multiple applications in the AWS Cloud. This eliminates the need to set up federation and identity provisioning with each application independently. This sharing feature also makes it easy to give your workforce (employees) access to many applications in different AWS accounts.

### Considerations for Sharing Identity Information in AWS Accounts

The attributes contained in AWS SSO are the basic attributes commonly used across applications. These attributes include information such as first and last name, phone number, email address, address, and preferred language. You might want to consider which applications and which accounts can use this personally identifiable information.

To control access to this information, you have two options. First, you can choose to enable access in only the AWS Organizations master account or in all AWS Organizations accounts. Second, you can use service control policies (SCPs) to control which applications can access the information in which AWS Organizations accounts. For example, if you enable access in the AWS Organizations master 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.

### Enable AWS SSO-Integrated Applications in AWS Accounts

When you enable AWS SSO for the first time, AWS enables use of integrated applications automatically in all AWS Organizations accounts. To constrain applications, you must implement SCPs.

If you enabled AWS SSO prior to November 25, 2019, AWS SSO disables the use of integrated applications in all AWS Organizations accounts. To use AWS SSO-integrated applications, you must
enable them in the master account and optionally enable them in member accounts. If you enable them in the master account only, you can enable them in member accounts in the future. To enable these applications, use the Enable access option in the AWS SSO Settings page in the AWS SSO-integrated applications section.

SAML Federation

AWS SSO supports identity federation with SAML (Security Assertion Markup Language) 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 consumer (called a service provider or SP). AWS SSO service uses this information to provide federated single sign-on (SSO) for those users who are authorized to use applications within the AWS SSO user portal.

AWS SSO adds SAML IdP capabilities to either your AWS Managed Microsoft AD or your AWS SSO store. Users can then SSO into services that support SAML, including the AWS Management Console and third-party applications such as Office 365, Concur, and Salesforce. At this time, AWS SSO does not support other directory types or IdPs.

User Authentications

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

Permission Sets

A permission set is a collection of administrator-defined policies that AWS SSO uses to determine a user’s effective permissions to access a given AWS account. Permission sets can contain either AWS managed policies or custom policies that are stored in AWS SSO. Policies are essentially documents that act as containers for one or more permission statements. These statements represent individual access controls (allow or deny) for various tasks that determine what tasks users can or cannot perform within the AWS account.

Permission sets are stored in AWS SSO and are only used for AWS accounts. They are not used to manage access to cloud applications. Permission sets ultimately get created as IAM roles in a given AWS account, with trust policies that allow users to assume the role through AWS SSO.
Manage Your Identity Source

You can configure your identity source in AWS SSO to determine where your users and groups are stored. Once configured, you can then look up users or groups in your store to grant them single sign-on access to AWS accounts, cloud applications, or both.

AWS SSO automatically provides you with a store by default, which you can use to manage your users and groups within AWS SSO. If you choose to store them in AWS SSO, create your users and groups and assign their level of access to your AWS accounts and applications. Alternatively, you can choose to Connect to Your External Identity Provider (p. 18) using Azure Active Directory, or Connect to Your Microsoft AD Directory (p. 14) using AWS Directory Service.

**Note**
AWS SSO does not support SAMBA4-based Simple AD as a connected directory.

**Topics**
- Considerations for Changing Your Identity Source (p. 9)
- Change Your Identity Source (p. 10)
- Manage Identities in AWS SSO (p. 10)
- Connect to Your Microsoft AD Directory (p. 14)
- Connect to Your External Identity Provider (p. 18)

Considerations for Changing Your Identity Source

Before you proceed with the steps in Change Your Identity Source (p. 10), we recommend that you first review the following important information. This information can help you understand how the process of switching your identity source might affect your current deployment.

**Changing Between AWS SSO and Active Directory**

Whether you switch from Microsoft AD to AWS SSO or from AWS SSO to Microsoft AD, the conversion deletes all users, groups, and assignments (entitlements). If you switch to AWS SSO, you must create your users, groups, and entitlements. If you switch to Microsoft AD, you must create entitlements with the users and groups that are in your Microsoft AD.

**Changing Between AWS SSO and Azure Active Directory**

If you switch to Azure AD, AWS SSO preserves all entitlements you had. The user and group entitlements will work if the AWS SSO user names and groups match those that you have in Azure AD. Any unmatched users and groups are unusable. If you switch from Azure AD to AWS SSO, AWS SSO preserves all users, groups, and entitlements. If any of the users previously had passwords in AWS SSO, then those users can continue signing in with their old passwords. The administrator must force a password reset for users that came from Azure AD and that did not previously exist in AWS SSO.

**Changing Between Microsoft AD and Azure Active Directory**

Whether you switch from Azure AD to Microsoft AD, or from Microsoft AD to Azure AD, the conversion deletes all users, groups, and assignments (entitlements) in AWS SSO. No user or group information is affected in either Azure AD or Microsoft AD. If you switch to Azure AD, you must configure AWS SSO to
provision your users. Or you must manually provision your Azure AD users and groups before you can configure entitlements. If you switch to Microsoft AD, you must create entitlements with the users and groups that are in your Microsoft AD.

**Change Your Identity Source**

You can change where you store users at any time. Use the following procedure to switch from a directory that AWS SSO provides (the default) to an external identity provider, AWS Managed Microsoft AD directory or vice versa. Make sure to review identity source considerations before proceeding. For more information, see Considerations for Changing Your Identity Source (p. 9).

**To change your identity source**

1. Open the AWS SSO console.
2. Choose Settings
4. On the Change identity source page, select the source you want to switch to, and then choose Next.
   - If you are switching to a Microsoft AD directory, you must choose the available directory from the provided menu.
   - **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.
5. Choose Next: Review.
6. Once you have read the disclaimer and are ready to proceed, type CONFIRM.
7. Choose Finish.

**Manage Identities in AWS SSO**

AWS Single Sign-On provides you with a default store where you can store your users and groups. If you choose to store them in AWS SSO, all you need to do is the following:

- 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.

If you prefer to manage users in AWS Managed Microsoft AD, you can discontinue use of your AWS SSO store at any time and instead connect AWS SSO to your Microsoft AD using AWS Directory Service. For more information, see Connect to Your Microsoft AD Directory (p. 14).

If you prefer to manage users in an external identity provider (IdP), you can connect AWS SSO to your IdP and enable automatic provisioning. For more information, see Connect to Your External Identity Provider (p. 18).

**Provisioning When Users are in AWS SSO**

When you create users and groups directly in AWS SSO, provisioning is automatic. These identities are immediately available for use in making assignments and for use by AWS SSO-integrated applications. For more information, see User and group provisioning (p. 6).
Add Users

Users and groups that you create in your AWS SSO store are available in AWS SSO only. Use the following procedure to add users to your AWS SSO store.

To add a user
1. Open the AWS SSO console.
2. Choose Users.
3. Choose Add user and provide the following required information:
   a. Username – This user name will be required to sign in to the user portal and cannot be changed later.
   b. Password – Choose from one of the following choices to send the user’s password.
      i. Send an email to the user with password setup instructions – This option automatically sends the user an email addressed from Amazon Web Services. The email invites the user on behalf of your company to access the AWS SSO user portal.
      ii. Generate a one-time password that you can share with the user – This option provides you with the user portal URL and password details that you can manually send to the user from your email address.
   c. Email address
   d. Confirm email address
   e. First name – You must enter a name here for automatic provisioning to work. For more information, see Automatic Provisioning (p. 20).
   f. Last name – You must enter a name here for automatic provisioning to work.
   g. Display name
      Note
      (Optional) You can provide additional attributes such as Employee ID and Office 365 Immutable ID to help map the user’s identity in AWS SSO with certain business applications that the user needs to use.
4. Choose Next: Groups.
5. Select one or more groups that you want the user to be a member of. Then choose Add user.

Add Groups

Use the following procedure to add groups to your AWS SSO store.

To add a group
1. Open the AWS SSO console.
2. Choose Groups.
3. Choose Create group.
4. In the Create group dialog box, enter a Group name and Description. The description should provide details on what permissions have been or will be assigned to the group.
5. Choose Create.

Add Users to Groups

Use the following procedure to add users as members of a group that you previously created in your AWS SSO store.

To add a user as a member of a group
1. Open the AWS SSO console.
2. Choose Groups.
3. Choose the group from the list.
4. On the group Details page, under Group members, choose Add users.
5. On the Add users to group page, locate the users you want to add as members. Then select the check box next to each of them.
6. Choose Add user.

Edit User Properties

Use the following procedure to edit the properties of a user in your AWS SSO store.

To edit user properties
1. Open the AWS SSO console.
2. Choose Users.
3. Choose the user that you want to edit.
4. On the user Details page, choose Edit user.
5. On the Edit user details page, make the updates to the properties as needed. Then choose Save changes.
   
   **Note**
   (Optional) You can modify additional attributes such as Employee ID and Office 365 Immutable ID to help map the user's identity in AWS SSO with certain business applications that users needs to use.

Disable a User

When you disable a user, you cannot edit their user details, reset their password, add the user to a group, or view their group membership. Use the following procedure to disable a user in your AWS SSO store.

To disable a user
1. Open the AWS SSO console.
2. Choose Users.
3. Choose the user you want to disable.
4. On the user Details page, choose Disable user.
5. On the **Disable user** dialog, choose **Disable user**.

   **Note**
   Disabling a user prevents them from being able to sign in to the user portal.

---

**Reset a User Password**

Use the following procedure to reset the password for a user in your AWS SSO store.

**To reset a user password**

1. Open the AWS SSO console.
2. Choose **Users**.
3. Choose 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.
   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.

---

**Password Requirements for the AWS SSO Identity Store**

When you create an AWS SSO identity store, a default password policy is created and applied to the store. Users who sign in to the user portal must adhere to the requirements in this policy when they set or change their password. The default password policy includes the following requirements:

- 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 (~!@#$%^&*_-+=`|(){}[]:;"'<>,.?/)  

   **Note**
   These requirements apply only to users created in the AWS SSO identity store. If you have configured an identity source other than AWS SSO 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 AWS SSO.

---

**Supported User and Group Attributes**

Attributes are pieces of information that help you define and identify individual user or group objects, such as `name`, `email`, or `members`. AWS SSO supports most commonly used attributes regardless if they are entered manually during user creation or when automatically provisioned using a synchronization engine such as defined in the System for Cross-Domain Identity Management (SCIM) specification. For
Connect to Your Microsoft AD Directory

AWS Single Sign-On enables administrators to connect their on-premises Active Directory (AD) or their AWS Managed Microsoft AD directory using AWS Directory Service. This Microsoft AD directory defines the pool of identities that administrators can pull from when using the AWS SSO console to assign single sign-on (SSO) access. After connecting their corporate directory to AWS SSO, administrators can then grant their AD users or groups access to AWS accounts, cloud applications, or both.

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

**Note**
AWS SSO does not support SAMBA4-based Simple AD as a connected directory.

Provisioning When Users Come from Active Directory

Because assigning permissions is a low-frequency activity, AWS SSO has the ability to access user information directly from Active Directory. AWS SSO connects to your directory through AWS Managed
Microsoft AD or AD Connector. For more information about how to do this, see Connect AWS SSO to an AWS Managed Microsoft AD Directory (p. 15).

However, AWS SSO-integrated applications require low-latency, high performance access to user information in the cloud. To accommodate this need, AWS SSO provisions user information into AWS SSO automatically each time a user signs in using just-in-time (JIT) provisioning. This process updates user information each time a user signs in so that attribute information is current. JIT provisioning is for users only.

If you delete a user from your Active Directory, the user can no longer sign-in to AWS SSO, AWS accounts, or any assigned applications. However, AWS SSO does not remove the provisioned user automatically. These JIT-provisioned users remain in AWS SSO and are visible to AWS SSO integrated applications until you remove them manually. Removing a JIT provisioned user has no effect in Active Directory. JIT provisioning is a one-way operation. Hence, if you accidentally remove a JIT provisioned user who is still in your Active Directory, AWS SSO reprovisions them automatically the next time they sign into AWS SSO. While removed, the user is unavailable for reference by AWS SSO-integrated applications.

For more information above provisioning, see User and group provisioning (p. 6).

Topics
- Connect AWS SSO to an AWS Managed Microsoft AD Directory (p. 15)
- Connect AWS SSO to an On-Premises Active Directory (p. 15)
- Attribute Mappings (p. 16)

**Connect AWS SSO to an AWS Managed Microsoft AD Directory**

Use the following procedure to connect an AWS Managed Microsoft AD directory that is managed by AWS Directory Service to AWS SSO.

**To connect AWS SSO to AWS Managed Microsoft AD**

1. Open the AWS SSO console.
   
   **Note**
   Make sure that the AWS SSO 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. Under **Identity source**, choose **Change**.
4. On the **Change identity source** page, choose **Active Directory**, choose the AWS Managed Microsoft AD directory from the list, and then choose **Next: Review**.
5. On the **Review and confirm** page, review the information and type **CONFIRM**.
6. Choose **Change identity source**.

**Connect AWS SSO to an On-Premises Active Directory**

Users in your on-premises Active Directory can also have SSO access to AWS accounts and cloud applications in the AWS SSO user portal. To do that, AWS Directory Service has the following two options available:
• **Create a two-way trust relationship** – When two-way trust relationships are created between AWS Managed Microsoft AD and an on-premises Active Directory, on-premises users can sign in with their corporate credentials to various AWS services and business applications. One-way trusts do not work with AWS SSO. For more information about setting up a two-way trust, see *When to Create a Trust Relationship* in the *AWS Directory Service Administration Guide*.

• **Create an AD Connector** – AD Connector is a directory gateway that can redirect directory requests to your on-premises Active Directory without caching any information in the cloud. For more information, see *Connect to a Directory* in the *AWS Directory Service Administration Guide*.

  **Note**
  If you are connecting AWS SSO to an AD Connector directory, any future user password resets must be done from within Active Directory. This means that users will not be able to reset their passwords from the user portal.

  **Note**
  AWS SSO does not work with SAMBA4-based Simple AD directories.

### Attribute Mappings

Attribute mappings are used to map attribute types that exist in AWS SSO with like attributes in an AWS Managed Microsoft AD directory. AWS SSO retrieves user attributes from your Microsoft AD directory and maps them to AWS SSO user attributes. These AWS SSO user attribute mappings are also used for generating SAML assertions for your cloud applications. Each cloud application determines the list of SAML attributes it needs for successful single sign-on.

AWS SSO prefills a set of attributes for you under the **Attribute mappings** tab found on your application's configuration page. AWS SSO uses these user attributes to populate SAML assertions (as SAML attributes) that are sent to the cloud application. These user attributes are in turn retrieved from your Microsoft AD directory. For more information, see *Map Attributes in Your Application to AWS SSO Attributes* (p. 41).

AWS SSO also manages a set of attributes for you under the **Attribute mappings** section of your directory configuration page. For more information, see *Map Attributes in AWS SSO to Attributes in Your AWS Managed Microsoft AD Directory* (p. 18).

### Supported Directory Attributes

The following table lists all AWS Managed Microsoft AD directory attributes that are supported and that can be mapped to user attributes in AWS SSO.

<table>
<thead>
<tr>
<th>Supported attributes in your Microsoft AD directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>${dir:email}</td>
</tr>
<tr>
<td>${dir:displayName}</td>
</tr>
<tr>
<td>${dir:distinguishedName}</td>
</tr>
<tr>
<td>${dir:firstName}</td>
</tr>
<tr>
<td>${dir:guid}</td>
</tr>
<tr>
<td>${dir:initials}</td>
</tr>
<tr>
<td>${dir:proxyAddresses}</td>
</tr>
<tr>
<td>${dir:lastname}</td>
</tr>
</tbody>
</table>
Supported attributes in your Microsoft AD directory

<table>
<thead>
<tr>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>${dir:proxyAddresses:smtp}</td>
</tr>
<tr>
<td>${dir:proxyAddresses:SMTP}</td>
</tr>
<tr>
<td>${dir:windowsUpn}</td>
</tr>
</tbody>
</table>

You can specify any combination of supported Microsoft AD directory attributes to map to a single attribute in AWS SSO. For example, you could choose the `preferredUsername` attribute under the **User attribute in AWS SSO** column. Then map it to either `${dir:displayname}` or `${dir:lastname}${dir:firstname }` or any single supported attribute or any arbitrary combination of supported attributes.

**Supported AWS SSO Attributes**

The following table lists all AWS SSO attributes that are supported and that can be mapped to user attributes in your AWS Managed Microsoft AD directory. Later, after you set up your application attribute mappings, you can use these same AWS SSO attributes to map to actual attributes used by that application.

<table>
<thead>
<tr>
<th>Supported attributes in AWS SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>${user:AD_GUID}</code></td>
</tr>
<tr>
<td><code>${user:email}</code></td>
</tr>
<tr>
<td><code>${user:familyName}</code></td>
</tr>
<tr>
<td><code>${user:firstName}</code></td>
</tr>
<tr>
<td><code>${user:middleName}</code></td>
</tr>
<tr>
<td><code>${user:name}</code></td>
</tr>
<tr>
<td><code>${user:preferredUsername}</code></td>
</tr>
<tr>
<td><code>${user:subject}</code></td>
</tr>
</tbody>
</table>

**Default Mappings**

The following table shows the default mappings for user attributes in AWS SSO to the user attributes in your AWS Managed Microsoft AD directory. At this time, AWS SSO only supports the list of attributes shown in the **User attribute in AWS SSO** column.

<table>
<thead>
<tr>
<th>User attribute in AWS SSO</th>
<th>Maps to this attribute in your Microsoft AD directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD_GUID</td>
<td><code>${dir:guid}</code></td>
</tr>
<tr>
<td>email</td>
<td><code>${dir:windowsUpn}</code></td>
</tr>
<tr>
<td>familyName</td>
<td><code>${dir:lastname}</code></td>
</tr>
<tr>
<td>givenName</td>
<td><code>${dir:firstname}</code></td>
</tr>
</tbody>
</table>
User attribute in AWS SSO | Maps to this attribute in your Microsoft AD directory
---|---
middleName | ${dir:initials}
name | ${dir:displayname}
pREFERREDUSERNAME | ${dir:displayname}
sUBJECT | ${dir:windowsUpn}

You can change the default mappings or add more attributes to the SAML assertion based on your requirements. For example, assume that your cloud application requires the users email in the User.Email SAML attribute. In addition, assume that email messages are stored in the windowsUpn attribute in your Microsoft AD directory. To achieve this mapping, you must make changes in the following two places in the AWS SSO console:

1. On the Directory page, under the Attribute mappings section, you would need to map the user attribute email to the `${dir:windowsUpn}` attribute (in the Maps to this attribute in your directory column).
2. On the Applications page, choose the application from the table. Choose the Attribute mappings tab. Then map the User.Email attribute to the `${user:email}` attribute (in the Maps to this string value or user attribute in AWS SSO column).

Please note that you must supply each directory attribute in the form `${dir:AttributeName}`. For example, the firstname attribute in your Microsoft AD directory becomes `${dir:firstname}`. It is important that every directory attribute have an actual value assigned. Attributes missing a value after `${dir:` will cause user sign-in issues.

### Map Attributes in AWS SSO to Attributes in Your AWS Managed Microsoft AD Directory

You can use the following procedure to specify how your user attributes in AWS SSO should map to corresponding attributes in your Microsoft AD directory.

**To map attributes in AWS SSO to attributes in your directory**

1. Open the AWS SSO console.
2. Choose Connected directory.
3. Under Attribute mappings, choose Edit attribute mappings.
4. On the Edit attribute mappings page, find the attribute in AWS SSO that you want to map and then type a value in the text box. For example, you might want to map the AWS SSO user attribute email to the Microsoft AD directory attribute `${dir:windowsUpn}`.
5. Choose Save changes.

### Connect to Your External Identity Provider

You can use AWS Single Sign-On (AWS SSO) to authenticate identities from external identity providers (IdPs) through the Security Assertion Markup Language (SAML) 2.0 standard. This enables your users to sign in to the AWS SSO user portal with their corporate credentials. They can then navigate to their assigned accounts, roles, and applications hosted in external identity providers.
For example, you can connect the Azure Active Directory (AD) IdP to AWS SSO. You can also assign access permissions centrally for their Azure AD users across all their AWS Organizations accounts, and applications. In addition, developers can simply sign in to the AWS Command Line Interface (CLI) using their Azure AD credentials and benefit from automatic short-term credential generation and rotation.

The SAML protocol does not provide a way to query the IdP to learn about users and groups. Therefore, you must make AWS SSO aware of those users and groups by provisioning them into AWS SSO.

**Provisioning When Users Come from an External Identity Provider**

When using an external IdP, you must provision all users and groups into AWS SSO before you can make any assignments to AWS accounts or applications. In this case you have two options: You can configure Automatic Provisioning (p. 20), or you can configure Manual Provisioning (p. 22) of your users and groups. Regardless of how you provision users, AWS SSO redirects the AWS Management Console, command line interface, and application authentication to your Azure AD. AWS SSO then grants access to those resources based on policies you create in AWS SSO. For more information about provisioning, see User and group provisioning (p. 6).

**How to Connect to an External Identity Provider**

Use the following procedure to connect to an external identity provider from the AWS SSO console.

**To connect to an external identity provider**

1. Open the AWS SSO console.
2. Choose **Settings**
3. On the **Settings** page, under **Identity source**, choose **Change**.
4. On the **Change identity source** page, select **External identity provider**. Then do the following:
   a. Under **Service provider metadata**, choose **Download metadata file** to download the metadata file and save it on your system. The AWS SSO SAML metadata file is required by your external identity provider.
   b. Under **Identity provider metadata**, choose **Browse** to search for the metadata file that you downloaded from your external identity provider. Then upload the file.
   c. Choose **Next: Review**.

**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.

5. Once you have read the disclaimer and are ready to proceed, type **CONFIRM**.
6. Choose **Finish**.

Once you have completed this procedure, you must configure Azure AD to use a nongallery application for AWS SSO. For more information, see Add an unlisted (non-gallery) application to your Azure AD organization on the Microsoft website.

**Topics**
- Automatic Provisioning (p. 20)
- Manual Provisioning (p. 22)
Automatic Provisioning

AWS SSO supports automatic provisioning (synchronization) of user and group information from Azure Active Directory (AD) into AWS SSO 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 AWS SSO. This causes the expected attributes to match between AWS SSO and your IdP. You configure this connection in your IdP using your SCIM endpoint for AWS SSO and a bearer token that you create in AWS SSO.

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 AWS SSO.

- It is imperative that your IdP provides a unique primary email address 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 make sure you configure SCIM in your IdP to map the non-Null unique email address to the AWS SSO primary email address attribute. And you must map the users non-Null unique sign-in identifier to the AWS SSO 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 AWS SSO primary email and the AWS SSO 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 AWS SSO once your identity provider sends those changes to AWS SSO. 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 AWS SSO 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 AWS SSO.

- 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 AWS SSO. To synchronize users and groups, make sure that they are assigned to the application or other setup that represents your IdP's connection to AWS SSO.

How to Enable Automatic Provisioning

Use the following procedure to enable automatic provisioning of users and groups from Azure AD to AWS SSO using the SCIM protocol.

**To automatically provision users from Azure AD to AWS SSO using the AWS SSO console**

1. In the AWS SSO console, choose **Settings** in the left navigation pane.
2. On the Settings page, under Identity source, choose Enable automatic provisioning. This immediately enables automatic provisioning in AWS SSO and displays the necessary endpoint and access token information.

3. In the Inbound automatic provisioning dialog box, copy or make note of the following values. You will need these later when you configure automatic provisioning in Azure AD.
   a. SCIM endpoint
   b. Configuration endpoint
   c. Access token

4. Choose Close.

Once you have completed this procedure, you must configure automatic provisioning in Azure AD. For more information, see Automate user provisioning and deprovisioning to applications with Azure Active Directory on the Microsoft website.

Important
Make sure that all users in Azure AD must have filled out both the First name and Last name in the user properties. Otherwise, automatic provisioning won’t work with Azure AD.

How to Disable Automatic Provisioning

Use the following procedure to disable automatic provisioning in the AWS SSO console.

Important
You must delete the access token before you start this procedure. For more information, see How to Delete an Access Token (p. 21).

To disable automatic provisioning in the AWS SSO console
1. In the AWS SSO console, choose Settings in the left navigation pane.
2. On the Settings page, under Identity source, choose View details.
3. In the Disable automatic provisioning dialog box, review the information and then choose Disable automatic provisioning.

How to Generate a New Access Token

Use the following procedure to generate a new access token in the AWS SSO console.

To generate a new access token
1. In the AWS SSO console, choose Settings in the left navigation pane.
3. In the Generate new access token dialog box, under Access token, choose Show token.

How to Delete an Access Token

Use the following procedure to delete an existing access token in the AWS SSO console.

To delete an existing access token
1. In the AWS SSO console, choose Settings in the left navigation pane.
2. On the Settings page, under Access tokens, near the access token you want to delete, choose Delete.
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 AWS SSO console. You add users to AWS SSO, making sure that you set the user name to be identical to the user name that you have in your IdP. At a minimum, you must have a unique email address and user name. For more information, see User name and email address uniqueness (p. 6).

You must also manage all groups manually in AWS SSO. To do this, you create the groups and add them using the AWS SSO console. For more information, see Groups (p. 6).
Manage SSO to Your AWS Accounts

AWS Single Sign-On is integrated with AWS Organizations so that administrators can pick multiple AWS accounts whose users need single sign-on (SSO) access to the AWS Management Console. These AWS accounts can be either the master account of the AWS Organizations or a member account. A master account is the AWS account that is used to create the organization. The rest of the accounts that belong to an organization are called member accounts. For more information about the different account types, see AWS Organizations Terminology and Concepts in the AWS Organizations User Guide.

Once you assign access from the AWS SSO console, you can use permission sets to further refine what users can do in the AWS Management Console. For more information about permission sets, see Permission Sets (p. 25).

Users follow a simple sign-in process:
1. Users use their directory credentials to sign in to the user portal.
2. Users then choose the AWS account name that will give them federated access to the AWS Management Console for that account.
3. Users who are assigned multiple permission sets choose which IAM role to use.

Permission sets are a way to define permissions centrally in AWS SSO so that they can be applied to all of your AWS accounts. These permission sets are provisioned to each AWS account as an IAM role. The user portal gives users the ability to retrieve temporary credentials for the IAM role of a given AWS account so they can use it for short-term access to the AWS CLI. For more information, see How to Get Credentials of an IAM Role for Use with CLI Access to an AWS Account (p. 44).

To use AWS SSO with AWS Organizations, you must first Enable AWS SSO (p. 3), which grants AWS SSO the capability to create Service-Linked Roles (p. 29) in each account in your AWS organization. These roles are not created until after you Assign User Access (p. 24) for a given account.

You can also connect an AWS account that is not part of your organization by setting up the account as a custom SAML application in AWS SSO. In this scenario, you provision and manage the IAM roles and trust relationships that are required to enable SSO access. For more information on how to do this, see Add and Configure a Custom SAML 2.0 Application (p. 35).

Topics
- Single Sign-On Access (p. 23)
- Permission Sets (p. 25)
- IAM Identity Provider (p. 28)
- Service-Linked Roles (p. 29)

Single Sign-On Access

You can assign users in your connected directory permissions to master or member AWS accounts in your AWS Organizations organization 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. AWS SSO configures all the necessary user permissions in your AWS accounts automatically.

Note
Only the IAM account root user or a user who has the AWSSOMasterAccountAdministrator IAM policy attached can grant users in your connected directory permissions to the master AWS account. For more information on how to delegate these permissions, see Delegate Who Can Assign SSO Access to Users in the Master Account (p. 25).
Assign User Access

Use the following procedure to assign SSO access to users and groups in your connected directory and use permission sets to determine their level of access.

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 access to users or groups

1. Open the AWS SSO console.

   Note
   Make sure that the AWS SSO console is using the Region where your AWS Managed Microsoft AD directory is located before you move to the next step.

2. Choose AWS accounts.

3. Under the AWS organization tab, in the list of AWS accounts, choose one or more accounts to which you want to assign access.

   Note
   The AWS SSO Console supports selecting up to 10 AWS accounts at a time per permission set when assigning user access. If you need to assign more than 10 AWS accounts to the same set of users, repeat this procedure for the additional accounts, selecting the same users and permission set when prompted.

4. On the AWS account details page, choose Assign users.

5. On the Select users or groups page, type a user or group name and choose Search connected directory. Once you have selected all the accounts that you want to assign access to, choose Next: Permission sets. You can specify multiple users or groups by selecting the applicable accounts as they appear in search results.

6. On the Select permission sets page, select the permission sets that you want to apply to the user or group from the table. Then choose Finish. You can optionally choose to Create a new permission set if none of the permissions in the table meets your needs. For detailed instructions, see Create Permission Set (p. 25).

7. Choose Finish to begin the process of configuring your AWS account.

   Note
   If this is the first time you have assigned SSO access to this AWS account, this process creates a service-linked role in the account. For more information, see Using Service-Linked Roles for AWS SSO (p. 63).

   Important
   The user assignment process may take a few minutes to complete. It is important that you leave this page open until the process successfully completes.

Remove User Access

Use this procedure when you need to remove SSO access to an AWS account for a particular user or group in your connected directory.

To remove user access from an AWS account

1. Open the AWS SSO console.
Delegate Who Can Assign SSO Access to Users in the Master Account

Assigning single sign-on access to the master account using the AWS SSO console is a privileged action. By default, only an AWS account root user, or a user who has the AWSSSOMasterAccountAdministrator AWS managed policy attached, can assign SSO access to the master account. The AWSSSOMasterAccountAdministrator provides manage SSO access to the master account within an AWS Organizations organization.

Use the following steps to delegate permissions to manage SSO access to users in your directory.

To grant permissions to manage SSO access to users in your directory

1. Sign in to the AWS SSO console as a root user of the master account or with another IAM user who has IAM administrator permissions to the master account.
2. Use the procedure Create Permission Set (p. 25) to create a permission set. When you get to step 5c, select the option Attach AWS managed policies. In the list of IAM policies that appear in the table, choose the AWSSSOMasterAccountAdministrator AWS managed policy. This policy grants permissions to any user who will be assigned access to this permission set in the future.
3. Use the procedure Assign User Access (p. 24) to assign the appropriate users to the permission set that you just created.
4. Communicate the following to the assigned users: When they sign in to the user portal and select the AWS Account icon, they must choose the appropriate IAM role name to be authenticated with the permissions that you just delegated.

Permission Sets

Permission sets define the level of access that users and groups have to an AWS account. Permission sets are stored in AWS SSO and provisioned to the AWS account as IAM roles. You can assign more than one permission set to a user. Users who have multiple permission sets must choose one when they sign in to the user portal. (Users will see these as IAM roles). For more information, see Permission Sets (p. 8).

Topics
- Create Permission Set (p. 25)
- Configure Permission Set Properties (p. 26)
- Delete Permission Sets (p. 28)

Create Permission Set

Use this procedure to create a permission set based on a custom permissions policy that you create, or on predefined AWS managed policies that exist in IAM, or both.

To create a permission set

1. Open the AWS SSO console.
2. Choose AWS accounts.
3. Select the Permission sets tab.
4. Choose Create permission set.
5. On the Create new permission set page, choose from one of the following options, and then follow the instructions provided under that option:

- **Use an existing job function policy**
  1. Under Select job function policy, select one of the default IAM job function policies in the list. For more information, see AWS Managed Policies for Job Functions.
  2. Choose Create.

- **Create a custom permission set**
  1. Under Create a custom permission set, type a name that will identify this permission set in AWS SSO. This name will also appear as an IAM role in the user portal for any users who have access to it.
  2. (Optional) You can also type a description. This description will only appear in the AWS SSO console and will not be visible to users in the user portal.
  3. (Optional) Specify the value for Session duration. This value is used to determine the length of time a user can be logged on before the console logs them out of their session. For more information, see Set Session Duration (p. 26).
  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 (p. 27).
  5. Select either Attach AWS managed policies or Create a custom permissions policy. Or select both if you need to link more than one policy type to this permission set.
  6. If you chose Attach AWS managed policies, under Attach AWS Managed policies, select up to 10 job-related or service-specific AWS managed policies from the list.
  7. If you chose Create a custom permissions policy, under Create a custom permissions policy, paste in a policy document with your preferred permissions. For a list of example policies to use for delegating AWS SSO tasks, see Customer Managed Policy Examples (p. 59).

For more information about the access policy language, see Overview of Policies in the IAM User Guide. To test the effects of this policy before applying your changes, use the IAM policy simulator.

8. Choose Create.

---

**Configure Permission Set Properties**

In AWS SSO you can customize the user experience by configuring the following permission set properties.

**Topics**

- Set Session Duration (p. 26)
- Set Relay State (p. 27)

**Set Session Duration**

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 has elapsed, AWS logs 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. AWS SSO 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 end-users federate into their AWS Account’s console or when using the AWS Command Line Interface (CLI), AWS SSO uses the session duration setting on the permission set to control the duration of the session. By default, AWS SSO-generated IAM roles for permission sets can only be assumed by SSO users, which ensures that the session duration specified in the SSO 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.

Once a permission set has been created, you can later update it to apply a new session duration. When you reapply the permission set to your AWS accounts, the IAM role’s maximum session duration value is updated. Use the following procedure to modify the session duration length for a given permission set.

**To set the session duration**

1. Open the AWS SSO console.
2. Choose AWS accounts.
3. Choose the Permission sets tab.
4. Choose the name of the permission set that will have the new session duration.
5. On the Permissions tab, next to Session duration, choose Edit.
6. On the Edit session duration page, next to New session duration, choose a new session length value, and then choose Continue.
7. Select the AWS accounts in the list that you want the new session duration value to apply to, and then choose Reapply permission set.

**Set Relay State**

During the federation authentication process, the relay state redirects users within the AWS Management Console. You can specify a relay state URL to redirect links to any service in the AWS Management Console. For example, the below illustration shows the process for redirecting to the S3 console (https://s3.console.aws.amazon.com/s3/home?region=us-east-1#).
Use the following procedure to modify the relay state URL for a given permission set.

**To set the relay state**

1. Open the AWS SSO console.
2. Choose AWS accounts.
3. Choose the Permission sets tab.
4. Choose the name of the permission set that will have the new relay state URL.
5. On the Permissions tab, choose Edit.
6. On the Edit general permission settings page, next to Relay state, type a URL value for any of the AWS services, and then choose Continue.
7. Select the AWS accounts in the list that you want the new relay state value to apply to, and then choose Reprovision.

**Delete Permission Sets**

Use this procedure to delete one or more permission sets so that they can no longer be used by any AWS account in the organization.

*Note*

All users and groups that have been assigned this permission set, regardless of what AWS account is using it, will no longer be able to sign in.

**To delete a permissions set from an AWS account**

1. Open the AWS SSO console.
2. Choose AWS accounts.
3. Choose the Permission sets tab.
4. Select the permission set you want to delete, and then choose Delete.
5. In the Delete permission set dialog box, choose Delete.

**IAM Identity Provider**

When you add SSO access to an AWS account, AWS SSO creates an IAM identity provider in each AWS account. An IAM identity provider helps keep your AWS account secure because you don't have to distribute or embed long-term security credentials, such as IAM access keys, in your application.

**Repair the IAM Identity Provider**

Use the following procedure to repair your identity provider in case it was deleted or modified.

**To repair an identity provider for an AWS account**

1. Open the AWS SSO console.
2. Choose AWS accounts.
3. In the table, select the AWS account that is associated with the identity provider that you want to repair.
4. On the AWS account details page, under IAM identity provider, choose Repair identity provider.
Remove the IAM Identity Provider

Use the following procedure to remove the IAM identity provider from AWS SSO.

**To remove the IAM identity provider from AWS SSO**

1. Open the AWS SSO management console
2. Choose **AWS accounts**
3. In the table select the AWS account that is associated with the IAM identity provider that you want to remove.
4. On the **Details** page for the AWS account, under **IAM identity provider**, choose **Remove identity provider**.

Service-Linked Roles

Service-linked roles are predefined IAM permissions that allow AWS SSO to delegate and enforce which users have SSO access to specific AWS accounts in your AWS organization. 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 AWS SSO to leverage those roles to perform service-related tasks. For more information, see AWS Organizations and Service-Linked Roles.

During the process to **Enable AWS SSO** (p. 3) for the first time, the AWS Organizations service grants AWS SSO the necessary permissions to create IAM roles in any of its AWS accounts. AWS SSO doesn't create roles in any of the AWS accounts at this point. It only creates a service-linked role in an AWS account after you have used the AWS SSO console to specify which account you want to assign SSO access to. For more information, see **Manage SSO to Your AWS Accounts** (p. 23).

Service-linked roles that are created in each AWS account are named `AWSServiceRoleForSSO`. For more information, see **Using Service-Linked Roles for AWS SSO** (p. 63).
Manage SSO to Your Applications

With AWS Single Sign-On, you can easily control who can have single sign-on (SSO) access to your cloud applications. Users get one-click access to these applications after they use their directory credentials to sign in to their user portal.

AWS SSO securely communicates with these applications through a trusted relationship between AWS SSO and the application's service provider. This trust is created when you add the application from the AWS SSO console and configure it with the appropriate metadata for both AWS SSO and the service provider.

After the application has been successfully added to the AWS SSO console, you can manage which users or groups need permissions to the application. By default, when you add an application, no users are assigned to the application. In other words, newly added applications in the AWS SSO console are inaccessible until you assign users to them. AWS SSO supports the following applications types:

- AWS SSO-integrated applications
- Cloud applications
- Custom Security Assertion Markup Language (SAML 2.0) applications

You can also grant your employees access to the AWS Management Console for a given AWS account in your organization. For more information on how to do this, see Manage SSO to Your AWS Accounts (p. 23).

The following sections explain how to configure user access to your AWS applications and third-party software as a service (SaaS) applications. You can also configure any custom applications that support identity federation with SAML 2.0.

Topics
- AWS SSO-Integrated Applications (p. 30)
- Cloud Applications (p. 32)
- Custom SAML 2.0 Applications (p. 35)
- Manage AWS SSO Certificates (p. 36)
- Application Properties (p. 38)
- Assign User Access (p. 40)
- Remove User Access (p. 40)
- Map Attributes in Your Application to AWS SSO Attributes (p. 41)

AWS SSO-Integrated Applications

With AWS SSO-Integrated Application Enablement (p. 7), AWS enterprise applications can exist in a child account in your organization but still use your AWS SSO identities. This provides your application end users with an easy sign-in experience and allows for delegation of administrator of your applications to operators in a child account.

Constraining AWS SSO-Integrated Application Use in AWS Accounts

If you want to constrain which of your AWS Organizations accounts that an integrated application can be used, you can do so using service control policies (SCPs). You can use SCPs to block access to the AWS
SSO user and group information and to prevent the application from being started except in designated accounts.

Add and Configure an AWS SSO-Integrated Application

To use AWS SSO-integrated applications, you must first enable AWS SSO to allow them access. For more information, see AWS SSO-Integrated Application Enablement (p. 7).

After they are enabled, integrated applications can access user and group information directly from AWS SSO. As a result, you won't have to manage access in both AWS SSO and then again inside the application. Instead, AWS SSO delegates application access to the application administrator. To add users to integrated applications, use the console of the application where you created the application.

Disable or Enable an AWS SSO-Integrated Application

If you only want to stop or restart user authentications to your application, use the following procedure to either disable or enable your application.

To disable or enable your application
1. In the AWS SSO console, choose Applications in the left navigation pane.
2. Select the application in the list.
3. Choose Actions, and then choose either Disable or Enable.

Remove an AWS SSO-Integrated Application

To remove an integrated application, visit the AWS Management Console where you manage your application. By removing the app this way, the application can gracefully remove resources that you might otherwise pay for. For emergency purposes only, you can force-remove an application from the AWS SSO console. AWS strongly recommends that you avoid this as such an action is irreversible and you might not be able to recover data from the application.

Before you force-remove, consider the following options:

- You can stop user authentications to this application without removing it by using the Disable option instead. For more information, see Disable or Enable an AWS SSO-Integrated Application (p. 31).
- If you want to disconnect the application from AWS SSO permanently, use the AWS Management Console where you created the application and remove it there instead. This helps to avoid unnecessary application-related charges that may otherwise appear if you continue with force-remove. This process also removes the application from AWS SSO.

Warning
Force-remove should only be used as a last resort. This operation deletes all user permissions to this application, disconnects the application from AWS SSO, and renders the application inaccessible.

To force-remove an AWS application
1. In the AWS SSO console, choose Applications in the left navigation pane.
2. Choose the application you want to remove in the list.
3. On the application Details page, under Remove Application, choose force-remove.
4. On the **Force-remove application** page, review the warning message. If you agree, enter `remove`, and then choose **Force-remove**.

## Cloud Applications

You can use the AWS SSO application configuration wizard to include built-in SAML integrations to many popular cloud applications. Examples include Salesforce, Box, and Office 365. For a complete list of applications that you can add from the wizard, see Supported Applications (p. 32).

Most cloud applications come with detailed instructions on how to set up the trust between AWS SSO and the application’s service provider. You can find these instructions on the cloud applications configuration page during the setup process and after the application has been set up. After the application has been configured, you can assign access to the groups or users that require it.

## Supported Applications

AWS SSO has built-in support for the following commonly used cloud applications.

**Note**
AWS Support engineers can assist customers who have Business and Enterprise support plans with some integration tasks that involve third-party software. For a current list of supported platforms and applications, see Third-Party Software Support on the AWS Support Features page.

<table>
<thead>
<tr>
<th>Application</th>
<th>Application</th>
<th>Application</th>
<th>Application</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000ft</td>
<td>Cybozu Mailwise</td>
<td>HappyFox</td>
<td>PhraseApp</td>
<td>Splunk Cloud</td>
</tr>
<tr>
<td>4me</td>
<td>Cybozu Office</td>
<td>Heap</td>
<td>PipeDrive</td>
<td>Splunk Enterprise</td>
</tr>
<tr>
<td>7Geese</td>
<td>Cybozu.com</td>
<td>HelloSign</td>
<td>Pivotal Tracker</td>
<td>Spotinst</td>
</tr>
<tr>
<td>Abstract</td>
<td>Dashlane</td>
<td>Helpdocs.io</td>
<td>PlanMyLeave</td>
<td>SproutVideo</td>
</tr>
<tr>
<td>Accredible</td>
<td>Databricks</td>
<td>HelpScout</td>
<td>PolicyIQ</td>
<td>Squadcast</td>
</tr>
<tr>
<td>Adobe Connect</td>
<td>Datadog</td>
<td>HighGear</td>
<td>ProcessPlan</td>
<td>Stackify</td>
</tr>
<tr>
<td>Adobe Creative Cloud</td>
<td>Declaree</td>
<td>Hightail</td>
<td>ProdPad</td>
<td>Status Hero</td>
</tr>
<tr>
<td>Adobe Sign</td>
<td>Deputy</td>
<td>Honey</td>
<td>Proto.io</td>
<td>StatusCast</td>
</tr>
<tr>
<td>Aha</td>
<td>DeskPro</td>
<td>Honeycomb.io</td>
<td>Proxyclick</td>
<td>StatusDashboard</td>
</tr>
<tr>
<td>Airbrake</td>
<td>Deskradar</td>
<td>HostedGraphite</td>
<td>PurelyHR</td>
<td>StatusHub</td>
</tr>
<tr>
<td>AlertOps</td>
<td>Detectify</td>
<td>HubSpot</td>
<td>Quip</td>
<td>Statuspage</td>
</tr>
<tr>
<td>AlertSite</td>
<td>Digicert</td>
<td>Humanity</td>
<td>Rapid7 Insight</td>
<td>StoriesOnBoard</td>
</tr>
<tr>
<td>Amazon Business</td>
<td>Dmarcian</td>
<td>IdeaScale</td>
<td>Recognize</td>
<td>Stormboard</td>
</tr>
<tr>
<td>Amazon WorkLink</td>
<td>Docebo</td>
<td>Igloo</td>
<td>Redash.io</td>
<td>SugarCRM</td>
</tr>
<tr>
<td>Andfrankly</td>
<td>DocuSign</td>
<td>Image Relay</td>
<td>Redlock</td>
<td>SumoLogic</td>
</tr>
<tr>
<td>AnswerHub</td>
<td>Dome9</td>
<td>iSpring</td>
<td>RescueAssist</td>
<td>SurveyGizmo</td>
</tr>
<tr>
<td>AppDynamics</td>
<td>Domo</td>
<td>IT Glue</td>
<td>RingCentral</td>
<td>SurveyMonkey</td>
</tr>
</tbody>
</table>

---

32
## Supported Applications

<table>
<thead>
<tr>
<th>AppFollow</th>
<th>Drift</th>
<th>JamaSoftware</th>
<th>Roadmunk</th>
<th>Syncplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area 1</td>
<td>Dropbox</td>
<td>Jamf</td>
<td>Robin</td>
<td>Tableau</td>
</tr>
<tr>
<td>Asana</td>
<td>DruvaInSync</td>
<td>Jenkins</td>
<td>Rollbar</td>
<td>Tableau Server</td>
</tr>
<tr>
<td>Assembla</td>
<td>Duo</td>
<td>JFrog Artifactory</td>
<td>Room Booking System</td>
<td>TalentLMS</td>
</tr>
<tr>
<td>Atlassian</td>
<td>Dynatrace</td>
<td>Jira</td>
<td>Salesforce</td>
<td>TargetProcess</td>
</tr>
<tr>
<td>Automox</td>
<td>EduBrite</td>
<td>Jitbit</td>
<td>Salesforce Service Cloud</td>
<td>TeamSupport</td>
</tr>
<tr>
<td>BambooHR</td>
<td>Egnyte</td>
<td>Jive</td>
<td>Samanage</td>
<td>Tenable.io</td>
</tr>
<tr>
<td>BenSelect</td>
<td>eLeaP</td>
<td>join.me</td>
<td>SAP BW</td>
<td>TextMagic</td>
</tr>
<tr>
<td>Bitabiz</td>
<td>Engagedly</td>
<td>Kanbanize</td>
<td>SAP Cloud Platform</td>
<td>ThousandEyes</td>
</tr>
<tr>
<td>Bitglass</td>
<td>Envoy</td>
<td>Keeper Security</td>
<td>SAP CRM ABAP</td>
<td>TinfoilSecurity</td>
</tr>
<tr>
<td>BlueJeans</td>
<td>Evernote</td>
<td>Kentik</td>
<td>SAP CRM Java</td>
<td>TitanFile</td>
</tr>
<tr>
<td>BMCRemedyforce</td>
<td>Expenseln</td>
<td>Kintone</td>
<td>SAP Enterprise Portal Java</td>
<td>TOPdesk Operator</td>
</tr>
<tr>
<td>Bonusly</td>
<td>Expensify</td>
<td>Klipfolio</td>
<td>SAP ERP ABAP</td>
<td>TOPdesk Self Service Desk</td>
</tr>
<tr>
<td>Box</td>
<td>Expiration Reminder</td>
<td>KnowledgeOwl</td>
<td>SAP EWM ABAP</td>
<td>Trakdesk</td>
</tr>
<tr>
<td>Brandfolder</td>
<td>External AWS Account</td>
<td>Kudos</td>
<td>SAP Fiori ABAP</td>
<td>Trello</td>
</tr>
<tr>
<td>Breezy HR</td>
<td>EZOfficelntventory</td>
<td>LiquidFiles</td>
<td>SAP GRC Access Control ABAP</td>
<td>Trend Micro Deep Security</td>
</tr>
<tr>
<td>Buddy Punch</td>
<td>EZRentOut</td>
<td>LiquidPlanner</td>
<td>SAP LMS</td>
<td>Uptime.com</td>
</tr>
<tr>
<td>Bugsee</td>
<td>Fastly</td>
<td>Litmos</td>
<td>SAP Netweaver ABAP</td>
<td>Uptrends</td>
</tr>
<tr>
<td>BugSnag</td>
<td>Federated Directory</td>
<td>LiveChat</td>
<td>SAP Netweaver Java</td>
<td>UserEcho</td>
</tr>
<tr>
<td>Buildkite</td>
<td>FileCloud</td>
<td>LogMeInRescue</td>
<td>SAP S4 ABAP</td>
<td>UserVoice</td>
</tr>
<tr>
<td>Bynder</td>
<td>FireHydrant</td>
<td>Lucidchart</td>
<td>SAP Solution Manager ABAP</td>
<td>Velpic</td>
</tr>
<tr>
<td>CakeHR</td>
<td>Fivetran</td>
<td>ManageEngine</td>
<td>SAP Solution Manager Java</td>
<td>Veracode</td>
</tr>
<tr>
<td>Canvas</td>
<td>Flock</td>
<td>MangoApps</td>
<td>SAP SRM ABAP</td>
<td>VictorOps</td>
</tr>
<tr>
<td>Chartio</td>
<td>FogBugz</td>
<td>Marketo</td>
<td>SAP xMII Java</td>
<td>vtiger</td>
</tr>
<tr>
<td>Chatwork</td>
<td>Formstack</td>
<td>Metricly</td>
<td>ScaleFT</td>
<td>WayWeDo</td>
</tr>
</tbody>
</table>
Add and Configure a Cloud Application

Use this procedure when you need to set up a SAML trust relationship between AWS SSO and your cloud application’s service provider. Before you begin this procedure, make sure you 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 it manually.

To add and configure a cloud application

1. In the AWS SSO console, choose **Applications** in the left navigation pane. Then choose **Add a new application**.
2. In the **Select an application** dialog box, select the application you want to add from the list. Then choose **Add**.
3. On the **Configure <application name>** page, under **Details**, enter a **Display name** for the application, such as **Salesforce**.
4. Under **AWS SSO metadata**, do the following:

<table>
<thead>
<tr>
<th>Circonus</th>
<th>Fossa</th>
<th>Miro</th>
<th>Scalyr</th>
<th>WeekDone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Webex</td>
<td>Freedcamp</td>
<td>MockFlow</td>
<td>ScreenSteps</td>
<td>WhosOnLocation</td>
</tr>
<tr>
<td>Cisco Meraki</td>
<td>Freshdesk</td>
<td>Mode Analytics</td>
<td>Seeit</td>
<td>Wordbee</td>
</tr>
<tr>
<td>Cisco Umbrella</td>
<td>FreshService</td>
<td>Moodle</td>
<td>Sentry.io</td>
<td>Workable</td>
</tr>
<tr>
<td>Citrix ShareFile</td>
<td>Front</td>
<td>MuleSoft Anypoint</td>
<td>ServiceNow</td>
<td>Workfront</td>
</tr>
<tr>
<td>Clarizen</td>
<td>G Suite</td>
<td>MyWebTimeSheets</td>
<td>SimpleMDM</td>
<td>Workplace by Facebook</td>
</tr>
<tr>
<td>ClickTime</td>
<td>GitBook</td>
<td>N2F Expense Reports</td>
<td>Skeddy</td>
<td>Workstars</td>
</tr>
<tr>
<td>Cloud CMS</td>
<td>Github</td>
<td>NewRelic</td>
<td>Skilljar</td>
<td>Wrike</td>
</tr>
<tr>
<td>Cloud Conformity</td>
<td>GitLab</td>
<td>Nuclino</td>
<td>Slack</td>
<td>xMatters</td>
</tr>
<tr>
<td>Cloud AMQP</td>
<td>Glasscubes</td>
<td>Office365</td>
<td>Slemma</td>
<td>XperienceHR</td>
</tr>
<tr>
<td>Cloud Checkr</td>
<td>GlassFrog</td>
<td>OnDMARC</td>
<td>Sli.do</td>
<td>Yodeck</td>
</tr>
<tr>
<td>Cloud Endure</td>
<td>GorillaStack</td>
<td>OpenVoice</td>
<td>Small Improvements</td>
<td>Zendesk</td>
</tr>
<tr>
<td>Cloud Passage</td>
<td>GoToAssist</td>
<td>OpsGenie</td>
<td>Smartsheet</td>
<td>Zephyr</td>
</tr>
<tr>
<td>CMNTY</td>
<td>GoTo Meeting</td>
<td>Pacific Timesheet</td>
<td>SnapEngage</td>
<td>Ziflow</td>
</tr>
<tr>
<td>Coder Pad</td>
<td>GoTo Training</td>
<td>PagerDuty</td>
<td>Snowflake</td>
<td>Zillable</td>
</tr>
<tr>
<td>Confluence</td>
<td>GoTo Webinar</td>
<td>Panopta</td>
<td>SonarQube</td>
<td>Zoho</td>
</tr>
<tr>
<td>Convo</td>
<td>Grovo</td>
<td>Panorama9</td>
<td>SparkPost</td>
<td>Zoho One</td>
</tr>
<tr>
<td>Coralogix</td>
<td>HackerOne</td>
<td>Park My Cloud</td>
<td>Spinnaker</td>
<td>Zoom</td>
</tr>
<tr>
<td>Cybozu Garoon</td>
<td>Hacker Rank</td>
<td>Peakon</td>
<td>Split.io</td>
<td></td>
</tr>
</tbody>
</table>

34
Custom SAML 2.0 Applications

You can use the AWS SSO application configuration wizard to add support for applications that allow identity federation using Security Assertion Markup Language (SAML) 2.0. In the console, you set these up by choosing Custom SAML 2.0 application from the application selector. Most of the steps for configuring a custom SAML application are the same as configuring a cloud application.

However, you also need to provide additional SAML attribute mappings for a custom SAML application. These mappings tell AWS SSO how to populate the SAML 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 attribute mappings on the application detail page that is accessible from the AWS SSO console.

Add and Configure a Custom SAML 2.0 Application

Use this procedure when you need to set up a SAML trust relationship between AWS SSO and your custom 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 add and configure a custom SAML application

1. In the AWS SSO console, choose Applications in the left navigation pane. Then choose Add a new application.
2. In the Select an application dialog box, select Custom SAML 2.0 application from the list. Then choose Configure application.
3. On the Configure <Custom app name> page, under Details, enter a Display name for the application, such as MyApp.
4. Under AWS SSO metadata, do the following:
   a. Next to AWS SSO SAML metadatafile, choose Download to download the identity provider metadata.
   b. Next to AWS SSO certificate, choose Download certificate 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.
5. (Optional) Under **Application properties**, you can specify additional properties for the **Application start URL**, **Relay State**, and **Session Duration**. For more information, see Application Properties (p. 38).

6. Under **Application metadata**, provide the **Application ACS URL** and **Application SAML audience** values.

7. Choose **Save changes** to save the configuration.

---

**Manage AWS SSO Certificates**

AWS SSO uses certificates to set up a SAML trust relationship between AWS SSO and your cloud application’s service provider. When you add an application in AWS SSO, an AWS SSO certificate is automatically created for use with that application during the setup process. By default, this autogenerated AWS SSO certificate is valid for a period of five years.

As an AWS SSO administrator, you’ll occasionally need to replace older certificates with newer ones for a given application. For example, you might need to replace a certificate when the expiration date on the certificate approaches. The process of replacing an older certificate with a newer one is referred to as **certificate rotation**.

**Topics**
- Considerations Before Rotating a Certificate (p. 36)
- Rotate an AWS SSO Certificate (p. 36)
- Certificate Expiration Status Indicators (p. 38)

**Considerations Before Rotating a Certificate**

Before you start the process of rotating a certificate in AWS SSO, consider the following:

- The certification rotation process requires that you reestablish the trust between AWS SSO and the service provider. To reestablish the trust, use the procedures provided in Rotate an AWS SSO Certificate (p. 36).
- 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 AWS SSO Certificate**

Rotating an AWS SSO 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 AWS SSO certificates that you generate can be configured to use the following properties:

- **Validity period** – Specifies the time allotted (in months) before a new AWS SSO 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 AWS SSO 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 AWS SSO 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 **AWS SSO metadata**, choose Manage certificate.
5. On the AWS SSO certificate page, choose Generate new certificate.
6. In the Generate new AWS SSO 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 AWS SSO 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. Whichever certificate is set as active, AWS SSO will use it to sign all SAML assertions.

1. Open the AWS SSO console.
2. Choose Applications.
3. In the list of applications, choose your application.
4. On the application details page, choose the **Configuration** tab. Under **AWS SSO metadata**, choose Manage certificate.
5. On the AWS SSO 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 **AWS SSO console**.  
2. Choose **Applications**.  
3. In the list of applications, choose your application.  
4. On the application details page, select the **Configuration** tab. Under **AWS SSO metadata**, choose **Manage certificate**.  
5. On the **AWS SSO certificate** page, select the certificate you want to delete. Choose **Actions** and then choose **Delete**.  
6. In the **Delete certificate** dialog box, choose **Delete**.

**Certificate Expiration Status Indicators**

While on the **Applications** page in the properties of an application, you may notice colored status indicator icons. These icons appear in the **Expires on** column next to each certificate in the list. The following describes the criteria that AWS SSO uses to determine which icon is displayed for each certificate.

- **Red** – Indicates that a certification is currently expired.  
- **Yellow** – Indicates that a certification will expire in 90 days or less.  
- **Green** – Indicates that a certification is currently valid and will remain valid for at least 90 more days.

**To check the current status of a certificate**

1. Open the **AWS SSO console**.  
2. Choose **Applications**.  
3. In the list of applications, review the status of the certificates in the list as indicated in the **Expires on** column.

**Application Properties**

In AWS SSO you can customize the user experience by configuring the following additional application properties.

**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 user 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 AWS SSO.  
3. AWS SSO 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, AWS SSO sends the relay state value along with a SAML response to the application.

Session Duration

Session duration is the length of time that the application user sessions are valid for. For SAML 2.0, this is used to set the `NotOnOrAfter` date of the SAML assertion's elements; `saml2:SubjectConfirmationData` and `saml2:Conditions`.

Session duration can be interpreted by applications in any of the following ways:

- Applications can use it to determine how long the SAML assertion is valid. Applications do not consider session duration when deciding the time allowed for the user.
- 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**

Use the following procedure to assign users SSO access to cloud applications or custom SAML 2.0 applications.

**Note**

- 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, 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. The user then automatically receives the permissions that are needed for the new organization.
- When assigning user access to applications, AWS SSO does not currently support users being added to nested groups. If a user is added to a nested group, they may receive a “You do not have any applications” message during sign-in. Assignments must be made against the immediate group the user is a member of.

**To assign access to users or groups**

1. Open the AWS SSO console.
   
   **Note**
   
   Make sure that the AWS SSO console is using the 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 an application to which you want to assign access.
4. On the application details page, choose the **Assigned users** tab. Then choose **Assign users**.
5. In the **Assign users** dialog box, enter a user or group name. Then choose **Search connected directory**. 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**

Use this procedure to remove user access to cloud applications or custom SAML 2.0 applications.

**To remove user access from an application**

1. Open the AWS SSO console.
2. Choose **Applications**.
3. In the list of applications, choose an application whose access you want to remove.
4. On the application details page, choose the **Assigned users** tab. Select the user or group that you want to remove and then choose **Remove**.
5. In the **Remove access** dialog box, verify the user or group name. Then choose **Remove access**.

## Map Attributes in Your Application to AWS SSO 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 AWS SSO.

**To map application attributes to attributes in AWS SSO**

1. Open the AWS SSO 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 **Attribute mappings** tab.
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 AWS SSO that you want to map to the application attribute. For example, you might want to map the application attribute **Username** to the AWS SSO user attribute **email**. To see the list of allowed user attributes in AWS SSO, see the table in [Attribute Mappings](#). (p. 16)
8. In the third column of the table, choose the appropriate format for the attribute from the menu.
9. Choose **Save changes**.
Using the User Portal

Your user portal provides you 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. From here you can quickly launch multiple applications simply by choosing the AWS account or application icon in the portal. The presence of icons in your portal means that an administrator or designated help desk employee 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 portal without additional sign-in prompts.

Contact your administrator or help desk to request additional access in the following situations:

- You don't see an AWS account or application that you need access to.
- The access that you have to a given account or application is not what you expected.

Topics

- Tips for Using the Portal (p. 42)
- How to Accept the Invitation to Join AWS SSO (p. 42)
- How to Sign In to the User Portal (p. 43)
- How to Sign Out of the User Portal (p. 43)
- How to Search for an AWS Account or Application (p. 43)
- How to Reset Your Password (p. 44)
- How to Get Credentials of an IAM Role for Use with CLI Access to an AWS Account (p. 44)
- How to Register a Device for Use with Multi-Factor Authentication (p. 45)

Tips for Using the Portal

Like any business tool or application that you use on a daily basis, the user portal might not work as you expected. If that happens, try these tips:

- Occasionally, you may need to sign out and sign back in to the user 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 user 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 user portal. Closing the application signs you out of that application only. Any other applications that you have opened from the user portal remain open and running.
- Before you can sign in as a different user, you must first sign out of the user portal. Signing out from the portal completely removes your credentials from the browser session.

How to Accept the Invitation to Join AWS SSO

If this is your first time signing into the user portal, check your email for instructions on how to activate your account.
To activate your account

1. Depending on the email you received from your company, choose one of the following methods to activate your account so that you can start using the user portal.
   a. If you received an email with the subject *Invitation to join AWS Single Sign-On*, open it and choose *Accept invitation*, which takes you to the *Single Sign-On* page. Here you specify a password, which you use each time you sign in to the portal. Once you have provided a password and have confirmed it, choose *Update User*.
   b. If you were sent an email from your company's IT support or IT administrator, follow the instructions they provided to activate your account.

2. Once you activate your account by providing a new password, the user portal signs you in automatically. If this does not occur, you can manually sign in to the user portal using the instructions provided in the next step.

How to Sign In to the User Portal

By this time, you should have been provided a specific sign-in URL to the user portal by an administrator or help desk employee. Once you have this, you can proceed with the following steps to sign in to the portal.

**Note**
Once you have been signed-in, your user portal session will be valid for 8 hours.

**To sign in to the user portal**

1. In your browser window, paste in the sign-in URL that you were provided. Then press Enter. 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 user name and password. 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 appears in the portal. Simply choose an icon.

How to Sign Out of the User Portal

When you sign out from the portal, your credentials are completely removed from the browser session.

**Note**
If you want to sign in as a different user, you must first sign out of the user portal.

**To sign out of the user portal**

- In the user portal, choose *Sign out* from the upper right corner of the portal.

How to Search for an AWS Account or Application

If your list of applications or AWS accounts is too large to find what you need, you can use the *Search* box.
How to Reset Your Password

From time to time you may need to reset your password, depending on your company policies.

To reset your password

1. Open a browser and go to the sign-in page for your user portal.
2. Under the Sign In button, choose Forgot Password?
3. Provide your Username and type the characters for the provided image to confirm that you are not a robot. Then choose Recover Password. This sends an email to you with the subject AWS Directory Service Reset Password Request.
4. Once you receive the email, choose Reset Password.
5. On the Single Sign-On page, need to specify a new password for the portal. Once you have provided a password and have confirmed it, choose Reset Password.

How to Get Credentials of an IAM Role for Use with CLI Access to an AWS Account

Use this procedure in the user portal when you need temporary security credentials for short-term access to resources in an AWS account using the AWS CLI. The user portal makes it easy for you to quickly select an AWS account and get the temporary credentials for a given IAM role. You can then copy the necessary CLI syntax (including all necessary credentials) and paste them into your AWS CLI command prompt.

By default, credentials retrieved through the user portal are valid for 1 hour. You can change this value up to 12 hours. Once you have completed this procedure, you can run any AWS CLI command (that your administrator has given you access to) until those temporary credentials have expired.

Note
Before you get started with the steps in this procedure, you must first install the AWS CLI. For instructions, see Installing the AWS Command Line Interface.

To get temporary credentials of an IAM role for CLI access to an AWS account

1. While signed into the portal, choose the AWS Accounts icon to expand the list of accounts.
2. Choose the AWS account from which you want to retrieve access credentials. Then, next to the IAM role name (for example Administrator), choose Command line or programmatic access.
3. In the Get credentials dialog box, choose either MacOS and Linux or Windows, depending on the operating system where you plan to use the CLI command prompt.
4. Depending on how you want to use the temporary credentials, choose one or more of the following options:
   - If you need to run commands from the AWS CLI in the selected AWS account, under Option 1: Set AWS environment variables, pause on the commands. Then choose Copy. Paste the commands into the CLI terminal window and press Enter to set the necessary environment variables.
   - If you need to run commands from multiple command prompts in the same AWS account, under Option 2: Add a profile to your AWS credentials file, pause on the commands. Then choose Copy. Paste the commands into your AWS credentials file to set up a newly named profile.
How to Register a Device for Use with Multi-Factor Authentication

Use the following procedure within the user portal to register your new device for multi-factor authentication (MFA).

**Note**
We recommend that you first download the appropriate Authenticator app onto your device before starting the steps in this procedure. For a list of apps that you can use for MFA devices, see Multi-Factor Authentication.

**To register your device for use with MFA**

1. Go to your user portal.
2. Near the top-right of the page, choose My devices.
3. On the My devices page, choose Register MFA device.

    **Note**
    If the Register MFA device option is grayed out, you will need to contact your administrator for assistance with registering your device.

4. On the Device name page, enter a friendly name for the new MFA device. It is helpful to describe the device to make it easy to identify and remove if your device is lost or stolen. For example, you might enter “My iPhone X.” Then choose Next. This name will be visible to your administrator.

5. The Device configuration page displays some information for the new MFA device, including an obscured QR code. Using the physical MFA device, do the following:

    1. Open a compatible MFA authenticator app. (For a list of apps that you can use for hosting MFA devices, see Multi-Factor Authentication.) If you are not sure which app to download, contact your administrator. If the MFA app supports multiple accounts (multiple MFA devices), choose the option to create a new account (a new MFA device).

    2. Determine whether the MFA app supports QR codes, and then do one of the following on the Device configuration page:

        a. Wait until no one is looking over your shoulder, choose Show QR code, and then use the app to scan the QR code.

        b. Wait until no one is looking over your shoulder, choose show secret key, and then enter that secret key into your MFA app.

6. On the Device configuration page, under An MFA code will be displayed on your device, Type that MFA code here., enter the one-time password that currently appears in the MFA app.

    **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 may become out of sync. This happens because time-based one-time passwords (TOTP) expire after a short period of time.
7. Choose Register MFA device. Your new MFA device can now start generating one-time passwords and is now ready for use with AWS.
Enable Multi-Factor Authentication

By default, when a user signs in to the user portal, they sign in with their email address and password (the first factor). This is the default authentication mechanism used in AWS SSO. But when multi-factor authentication (MFA) is enabled, users enter an MFA code (the second factor) that is generated by an application on their phone. Users must use this MFA code to be authenticated to the user portal. These factors together provide additional security by preventing access to your AWS organization unless users supply valid user credentials and a valid MFA code.

Topics

• Considerations Before Using MFA in AWS SSO (p. 47)
• Authentication Methods (p. 47)
• MFA Device Enforcement (p. 48)
• RADIUS MFA (p. 49)
• Authenticator Applications on User Devices (p. 49)
• How to Enable MFA (p. 50)
• How to Disable MFA (p. 50)
• How to Register an MFA Device (p. 50)
• How to Allow Users to Register Their Own MFA Devices (p. 51)

Considerations Before Using MFA in AWS SSO

Before you enable MFA, consider the following information:

• All users must have access to a physical device that can have applications installed on it, like a smartphone or tablet. Such a device is required before users can sign in using MFA. Therefore, you will need either to provide a device to each user or send them instructions on how they can register their own personal devices. For more information, see Authenticator Applications on User Devices (p. 49).
• Do not use the option Require Them to Provide a One-Time Password Sent by Email if your users must sign in to the user portal to access their email. For example, your users might use Office 365 on the user portal to read their email. In this case, users would not be able to retrieve the verification code and would be unable to sign in to the user portal. For more information, see Require Them to Provide a One-Time Password Sent by Email (p. 49).
• If you are already using RADIUS MFA that you configured with AWS Directory Service, then you do not need to enable MFA within AWS SSO. MFA is an alternative to RADIUS MFA for Microsoft Active Directory users of AWS SSO. For more information, see RADIUS MFA (p. 49).

Authentication Methods

Authentication methods help you determine the level of security that you want to enforce across all your users during sign-in. MFA has the following methods available:

• Context-aware
• Always-on
• Disabled
Note
You can configure AWS SSO to use a connected directory and decide to choose either the Context-aware or Always-on option. In these cases, your users must sign in to the user portal using the down-level logon name format (DOMAIN\UserName). This restriction does not apply when you are using an AWS SSO store. With an AWS SSO store, users can sign in using either their down-level logon name format or their UPN logon name format (UserName@Corp.Example.com). For general information about sign-in formats, see User Name Formats on the Microsoft documentation website.

Context-Aware

Context-aware is the default setting when you first configure AWS SSO. In this mode, AWS SSO analyzes the sign-in context (browser, location, and devices) for each user. AWS SSO then determines whether the user is signing in with a previously trusted context. If a user is signing in from an unknown IP address or is using an unknown device, SSO prompts the user for multi-factor authentication. The user is prompted for an MFA code in addition to their email address and password credentials.

This mode provides additional protection for users who frequently sign in from their offices. This mode is also easier for those users because they do not need to complete MFA on every sign-in. SSO prompts users with MFA once and permits them to trust their device. Once a user indicates that they want to trust a device, AWS SSO considers future sign-ins to be “trusted.” AWS SSO does not challenge the user for an MFA code when they use that trusted device. Users are only required to provide additional verification when their sign-in context changes. Such changes include signing in from a new device, a new browser, or an unknown IP address.

Note
Changing from Disabled mode to Context-aware mode overrides existing RADIUS MFA settings that are configured in AWS Directory Service for sign-in to AWS SSO for this directory. For more information, see RADIUS MFA (p. 49).

Always-On

In this mode, AWS SSO requires that users who have registered an MFA device provide an MFA code on every sign-in. You should use this mode if you have organizational or compliance policies that require your users to complete MFA every time they sign in to the user portal. For example, PCI DSS strongly recommends MFA during every sign-in to access applications that support high-risk payment transactions.

Disabled

While in this mode, no MFA authentication method is enabled. Users continue to sign in using their user name, password and/or RADIUS MFA as normal.

MFA Device Enforcement

The following options can be used to determine whether your users must have a registered MFA device when signing in to the user portal. These options also determine the method by which your users will receive their MFA code.

Allow Them to Sign In

Allow them to sign in is the default setting when you first configure AWS SSO MFA. Use this option to indicate that MFA devices are not required in order for your users to sign in to the user portal. Users who chose to register MFA devices will still be prompted for MFA codes.
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 an MFA code 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.

Require Them to Provide a One-Time Password Sent by Email

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 SSO enabled Identity source, the email address used will always be based on the AD 'email' attribute. Custom AD attribute mappings will not override this behavior.

RADIUS MFA

Remote Authentication Dial-In User Service (RADIUS) is an industry-standard client-server protocol that provides authentication, authorization, and accounting management so users can connect to network services. AWS Directory Service includes a RADIUS client that connects to the RADIUS server upon which you have implemented your MFA solution. For more information, see Enable Multi-Factor Authentication for AWS Managed Microsoft AD.

You can use either RADIUS MFA or MFA in AWS SSO for user sign-ins to the user portal, but not both. MFA in AWS SSO 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 AWS SSO, your users need an MFA code to sign in to the AWS SSO user portal. If you had previously used RADIUS MFA, enabling MFA in AWS SSO effectively overrides RADIUS MFA for users who sign in to the user portal. However, RADIUS MFA continues to challenge users when they sign in to all other applications that work with AWS Directory Service, such as Amazon WorkDocs.

If your MFA is Disabled on the AWS SSO console and you have configured RADIUS MFA with AWS Directory Service, RADIUS MFA governs user portal sign-in. This means that AWS SSO falls back to RADIUS MFA configuration if MFA is disabled.

Authenticator Applications on User Devices

Your users can use their internet accessible devices, such as a smartphone or tablet, as an MFA device. To do this, users must install an AWS supported mobile app that generates a six-digit authentication code.
Because these apps can run on unsecured mobile devices, MFA might not provide the same level of security as U2F devices or hardware MFA devices. You can enable only two MFA devices per user.

For a list of MFA apps that you can use on smartphones or tablets, see Multi-Factor Authentication. Note that AWS requires an MFA app that produces a six-digit one-time password.

How to Enable MFA

Use the following procedure to enable MFA in the AWS SSO console. Before you enable MFA, we recommend that you first review details about Authentication Methods (p. 47).

To enable MFA
1. Open the AWS SSO console.
2. In the left navigation pane, choose Settings.
3. On the Settings page, choose Configure.
4. On the Configure authentication page, choose one of the following Authentication Methods (p. 47) based on your business needs:
   - Only when their sign-in context changes (context-aware)
   - Every time they sign in (always-on)
5. Choose Save Changes.

How to Disable MFA

Use the following procedure to disable MFA in the AWS SSO console.

To disable MFA
1. Open the AWS SSO console.
2. In the left navigation pane, choose Settings.
3. On the Settings page, choose Configure.
4. On the Configure authentication page, choose Never (disabled).
5. Choose Save Changes.

How to Register an MFA Device

Use the following procedure to set up a new MFA device for access by a specific user in the AWS SSO console. You must have physical access to the user's MFA device in order to register it. For example, you might configure MFA for a user who will use an MFA device running on a smartphone. In that case, you must have the smartphone available in order to finish the wizard. Because of this, you might want to let users configure and manage their own MFA devices. For details on how to set this up, see How to Allow Users to Register Their Own MFA Devices (p. 51).

To register an MFA device
1. Open the AWS SSO console.
2. In the left navigation pane, choose Directory.
3. On the **Users** tab, choose a user in the list.
4. On the user's **Details** page, under **Multi-factor authentication (MFA) devices**, choose **Register MFA device**.
5. On the **Device name** page, type a friendly name for the new MFA device, and then choose **Next**. If you have enabled the option to allow users to manage their own devices, this user will see this friendly name in the user portal.
6. On the **Device configuration** page, AWS SSO 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.
7. Using the physical MFA device, do the following:
   1. Open a compatible MFA authenticator app. (For a list of apps that you can use for hosting MFA devices, see Multi-Factor Authentication.) If the MFA app supports multiple accounts (multiple MFA devices), choose the option to create a new account (a new MFA device).
   2. Determine whether the MFA app supports QR codes, and then do one of the following on the **Device configuration** page:
      a. 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.
      b. Choose **show secret key**, and then type that secret key into your MFA app.
      **Important**
      When you configure an MFA device for AWS SSO, 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 AWS SSO for the user.
8. On the **Device configuration** page, under **Type the MFA code generated by the app**, 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.
9. Choose **Register MFA device**. The MFA device can now start generating one-time passwords and is now ready for use with AWS.

## How to Allow Users to Register Their Own MFA Devices

*Use the following procedure to allow your users to self-register their own MFA devices.*

### To allow users to register their own MFA devices

1. Open the **AWS SSO console**.
2. In the left navigation pane, choose **Settings**.
3. On the **Settings** page, choose **Configure**.
4. On the **Configure authentication** page, under **Who can manage MFA devices**, choose **Users and administrators can add and manage MFA devices**.
5. Choose **Save Changes**.
Note
After you set up self-registration for your users, you may want to send them a link to the procedure How to Register a Device for Use with Multi-Factor Authentication (p. 45). This topic provides instructions on how to set up their own MFA devices.
Security in AWS Single Sign-On

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 Single Sign-On, 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 AWS SSO. The following topics help you how to configure AWS SSO to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS SSO resources.

**Topics**

- Identity and Access Management for AWS SSO (p. 53)
- Logging and Monitoring in AWS Single Sign-On (p. 65)
- Compliance Validation for AWS Single Sign-On (p. 69)
- Resilience in AWS Single Sign-On (p. 70)
- Infrastructure Security in AWS Single Sign-On (p. 70)

Identity and Access Management for AWS SSO

Access to AWS SSO requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an AWS SSO application.

Authentication to the AWS SSO user portal is controlled by the directory that you have connected to AWS SSO. However, authorization to the AWS accounts that are available to end users from within the user portal is determined by two factors:

1. Who has been assigned access to those AWS accounts in the AWS SSO console. For more information, see Single Sign-On Access (p. 23).
2. What level of permissions have been granted to the end users in the AWS SSO console to allow them the appropriate access to those AWS accounts. For more information, see Permission Sets (p. 25).

The following sections explain how you as an administrator can control access to the AWS SSO console or can delegate administrative access for day-to-day tasks from the AWS SSO console.

- Authentication (p. 54)
- Access Control (p. 55)
Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a directory in AWS SSO). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS SSO supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.

- **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.
Access Control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access AWS SSO resources. For example, you must have permissions to create an AWS SSO connected directory.

The following sections describe how to manage permissions for AWS SSO. We recommend that you read the overview first.

- Overview of Managing Access Permissions to Your AWS SSO Resources (p. 55)
- Using Identity-Based Policies (IAM Policies) for AWS SSO (p. 58)
- Using Service-Linked Roles for AWS SSO (p. 63)

Overview of Managing Access Permissions to Your AWS SSO Resources

Every AWS resource is owned by an AWS account, and permissions to create or access the resources are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support attaching permissions policies 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.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

- AWS SSO Resources and Operations (p. 55)
- Understanding Resource Ownership (p. 55)
- Managing Access to Resources (p. 56)
- Specifying Policy Elements: Actions, Effects, Resources, and Principals (p. 57)
- Specifying Conditions in a Policy (p. 57)

AWS SSO Resources and Operations

In AWS SSO, 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, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:
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 AWS SSO. 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. AWS SSO supports only identity-based policies (IAM policies).

**Topics**
- Identity-Based Policies (IAM Policies) (p. 56)
- Resource-Based Policies (p. 57)

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

**Attach a permissions policy to a user or a group in your 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 AWS SSO 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 example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions to resources in Account A.
2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

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 AWS SSO 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 AWS SSO resources that are owned by the account.

```
{
   "Version":"2012-10-17",
   "Statement“:[
   {
      "Effect":"Allow",
      "Action":"sso:List*",
      "Resource":"*"
   }
   ]
}
```

For more information about using identity-based policies with AWS SSO, see Using Identity-Based Policies (IAM Policies) for AWS SSO (p. 58). 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. AWS SSO doesn’t support resource-based policies.

**Specifying Policy Elements: Actions, Effects, Resources, and Principals**

For each AWS SSO resource (see AWS SSO Resources and Operations (p. 55)), the service defines a set of API operations. To grant permissions for these API operations, AWS SSO 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. For AWS SSO resources, you always use the wildcard character (*) in IAM policies. For more information, see AWS SSO Resources and Operations (p. 55).
- **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 AWS SSO `DescribePermissionsPolicies` operation.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don’t explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS SSO 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 AWS SSO. 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.

Using Identity-Based Policies (IAM Policies) for AWS SSO

This topic provides examples of permissions policies that an account administrator can attach to IAM identities (that is, users, groups, and roles).

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 AWS SSO resources. For more information, see Overview of Managing Access Permissions to Your AWS SSO Resources (p. 55).

The sections in this topic cover the following:

• Permissions Required to Use the AWS SSO Console (p. 59)
• AWS Managed (Predefined) Policies for AWS SSO (p. 59)
• Customer Managed Policy Examples (p. 59)

The following shows an example of a permissions policy.

```json
{
"Version" : "2012-10-17",
"Statement" : [
{
"Action" : [
"sso:CreateApplicationInstance",
"sso:UpdateResponseConfig",
"sso:UpdateResponseSchemaConfig",
"sso:UpdateSecurityConfig",
"sso:UpdateServiceProviderConfig",
"sso:UpdateApplicationInstanceStatus",
"sso:UpdateApplicationInstanceDisplay",
"sso:CreateProfile",
"sso:SetupTrust"
],
"Effect" : "Allow",
"Resource" : "*"
},
{
"Action" : [
"organizations:xxx",
"organizations:yyy"
],
"Effect" : "Allow",
"Resource" : "*"
},
{
"Action" : [
"ds:AuthorizeApplication"
],
"Effect" : "Allow",
"Resource" : "*"
}
]}
```
The policy includes the following:

- The first statement grants permission to manage profile associations to users and groups within your directory. It also grants permission to read all of the AWS SSO resources.
- The second statement grants permissions to search the directory for users and groups. This is required before you can create profile associations.

The policy doesn't specify the Principal element because in an identity-based policy you don't specify the principal who gets the permission. When you attach a policy to a user, the user is the implicit principal. When you attach a permission policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

Permissions Required to Use the AWS SSO Console

For a user to work with the AWS SSO console, that user must have permissions listed in the preceding policy.

If you create an IAM policy that is more restrictive than the minimum required permissions, the console won't function as intended for users with that IAM policy.

AWS Managed (Predefined) Policies for AWS SSO

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the IAM User Guide.

Customer Managed Policy Examples

In this section, you can find example user policies that grant permissions for various AWS SSO actions.

Examples

- Example 1: Allow a User to Set Up and Enable AWS SSO (p. 59)
- Example 2: Allow a User to Manage Your AWS SSO Connected Directory (p. 60)
- Example 3: Allow a User to Manage Applications in AWS SSO (p. 60)
- Example 4: Allow a User to Manage Permissions for Your AWS Accounts in AWS SSO (p. 61)
- Example 5: Allow a User to Manage Access for Your Applications in AWS SSO (p. 62)
- Example 6: Allow a User to Find Which Cloud Applications Are Preintegrated with AWS SSO (p. 62)
- Example 7: Allow a User to Add Users and Groups in AWS SSO (p. 63)

Example 1: Allow a User to Set Up and Enable AWS SSO

The following permissions policy grants permissions to allow a user to open the AWS SSO console and enable the service. In order to do so, permissions such as those granted to the AWS Organizations master account are also required.

```json
{
   "Version":"2012-10-17",
   "Statement":[
```
Example 2: Allow a User to Manage Your AWS SSO Connected Directory

The following permissions policy grants permissions to a user to manage your connected directory.

```json
{
"Version":"2012-10-17",
"Statement":[
{
"Effect":"Allow",
"Action": [
"sso:AssociateDirectory",
"sso:DisassociateDirectory",
"sso:ListDirectoryAssociations",
"sso:UpdateDirectoryAssociation"
],
"Resource": "*"
}
]
}
```

Example 3: Allow a User to Manage Applications in AWS SSO

The following permissions policy grants permissions to allow a user to create and manage application instances, profiles, and certificates in the AWS SSO console.

```json
{
"Version":"2012-10-17",
"Statement":[
{
"Effect":"Allow",
"Action": [
"sso:ListApplicationTemplates",
"sso:GetApplicationTemplate",
"sso:ListApplicationInstances",
"sso:GetApplicationInstance"
]
}
]
```
Example 4: Allow a User to Manage Permissions for Your AWS Accounts in AWS SSO

The following permissions policy grants permissions to allow a user to create and manage permission sets for your AWS accounts in the AWS SSO console.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:ListApplicationInstances",
        "sso:GetApplicationInstance",
        "sso:CreateApplicationInstance",
        "sso:UpdateApplicationInstanceStatus",
        "sso:UpdateApplicationInstanceDisplayData",
        "sso:UpdateApplicationInstanceServiceProviderConfiguration",
        "sso:UpdateApplicationInstanceResponseConfiguration",
        "sso:UpdateApplicationInstanceResponseSchemaConfiguration",
        "sso:UpdateApplicationInstanceSecurityConfiguration",
        "sso:DeleteApplicationInstance",
        "sso:ImportApplicationInstanceServiceProviderMetadata",
        "sso:CreateProfile",
        "sso:UpdateProfile",
        "sso:DeleteProfile",
        "sso:GetProfile",
        "sso:ListProfiles",
        "sso:ListApplicationInstanceCertificates",
        "sso:CreateApplicationInstanceCertificate",
        "sso:UpdateApplicationInstanceActiveCertificate",
        "sso:DeleteApplicationInstanceCertificate",
        "sso:CreatePermissionSet",
        "sso:GetPermissionSet",
        "sso:ListPermissionSets",
        "sso:DeletePermissionSet",
        "sso:PutPermissionsPolicy",
        "sso:DeletePermissionsPolicy",
        "sso:DescribePermissionsPolicies",
        "sso:GetTrust",
```
"sso:CreateTrust",
"sso:UpdateTrust",
"sso:DeleteTrust"
],

"Resource": "*",
}

Example 5: Allow a User to Manage Access for Your Applications in AWS SSO

The following permissions policy grants permissions to allow a user to manage who can access your applications in the AWS SSO console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:ListApplicationInstances",
        "sso:ListProfileAssociations",
        "sso:AssociateProfile",
        "sso:DisassociateProfile"
      ],
      "Resource": "*",
    },
    {
      "Effect": "Allow",
      "Action": [
        "ds:DescribeDirectories"
      ],
      "Resource": "*",
    }
  ]
}
```

Example 6: Allow a User to Find Which Cloud Applications Are Preintegrated with AWS SSO

The following permissions policy grants permissions to allow a user to locate what cloud applications are preintegrated with AWS SSO using the Add Application wizard.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso:ListApplicationTemplates",
        "sso:GetApplicationTemplate"
      ],
      "Resource": "*",
    }
  ]
}
```
Example 7: Allow a User to Add Users and Groups in AWS SSO

The following permissions policy grants permissions to allow a user to open the AWS SSO console and add users and groups in the directory that AWS SSO provides by default.

```json
{
    "Version":"2012-10-17",
    "Statement": [
        {  
            "Effect":"Allow",
            "Action": [ 
                "sso-directory:*"
            ],
            "Resource": "*"
        }
    ]
}
```

Using Service-Linked Roles for AWS SSO

AWS Single Sign-On 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 AWS SSO. It is predefined by AWS SSO and includes all the permissions that the service requires to call other AWS services on your behalf. For more information, see Service-Linked Roles (p. 29).

A service-linked role makes setting up AWS SSO easier because you don’t have to manually add the necessary permissions. AWS SSO defines the permissions of its service-linked role, and unless defined otherwise, only AWS SSO 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 AWS SSO

AWS SSO uses the service-linked role named AWSServiceRoleForSSO to grant AWS SSO 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:

* AWS SSO

The AWSServiceRoleForSSO service-linked role permissions policy allows AWS SSO 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:DeleteRolePolicy
* iam:DetachRolePolicy
* iam:GetRole
* iam:ListRolePolicies
• `iam:PutRolePolicy`
• `iam:ListAttachedRolePolicies`

The `AWS::ServiceRoleForSSO` service-linked role permissions policy allows AWS SSO to complete the following on SAML providers with name prefix as "AWSSSO_":

• `iam:CreateSAMLProvider`
• `iam:GetSAMLProvider`
• `iam:UpdateSAMLProvider`
• `iam:DeleteSAMLProvider`

The `AWS::ServiceRoleForSSO` service-linked role permissions policy allows AWS SSO to complete the following on all organizations:

• `organizations:DescribeAccount`
• `organizations:DescribeOrganization`
• `organizations:ListAccounts`

The `AWS::ServiceRoleForSSO` service-linked role permissions policy allows AWS SSO to complete the following on all IAM roles (*):

• `iam:listRoles`

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 AWS SSO**

You don't need to manually create a service-linked role. When a user who is signed in with the AWS organization’s master account assigns access to an AWS account for the first time, AWS SSO creates the service-linked role automatically in that AWS account.

**Important**

If you were using the AWS SSO service before December 7, 2017, when it began supporting service-linked roles, then AWS SSO created the `AWS::ServiceRoleForSSO` 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 AWS SSO**

AWS SSO does not allow you to edit the `AWS::ServiceRoleForSSO` 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 AWS SSO

You don't need to manually delete the AWSServiceRoleForSSO role. When an AWS account is removed from an AWS organization, AWS SSO 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 AWS SSO 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 AWS SSO resources used by the AWSServiceRoleForSSO

1. Remove User Access (p. 24) for all users and groups that have access to the AWS account.
2. Delete Permission Sets (p. 28) that you have associated with the AWS account.
3. Remove the IAM Identity Provider (p. 29) to delete the trust between AWS SSO and 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.

Logging and Monitoring in AWS Single Sign-On

As a best practice, you should monitor your organization to ensure that changes are logged. This helps you to ensure that any unexpected change can be investigated and unwanted changes can be rolled back. AWS Single Sign-On currently supports two AWS services that help you monitor your organization and the activity that happens within it.

Topics
- Logging AWS SSO API Calls with AWS CloudTrail (p. 65)
- Amazon CloudWatch Events (p. 69)

Logging AWS SSO API Calls with AWS CloudTrail

AWS SSO is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS SSO. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, Amazon CloudWatch Logs, and Amazon CloudWatch Events. Using the information collected by CloudTrail, you can determine the request that was made to AWS SSO, 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.

AWS SSO Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS SSO, that activity is recorded in a CloudTrail event along with other AWS service events in Event history.
You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS SSO, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

When CloudTrail logging is enabled in your AWS account, API calls made to AWS SSO actions are tracked in log files. AWS SSO records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

The following AWS SSO CloudTrail actions are supported:

- AssociateDirectory
- AssociateProfile
- CreateApplicationInstance
- CreateApplicationInstanceCertificate
- CreatePermissionSet
- CreateProfile
- DeleteApplicationInstance
- DeleteApplicationInstanceCertificate
- DeletePermissionsPolicy
- DeletePermissionSet
- DeleteProfile
- DescribePermissionsPolicies
- DisassociateDirectory
- DisassociateProfile
- GetApplicationInstance
- GetApplicationTemplate
- GetMfaDeviceManagementForDirectory
- GetPermissionSet
- GetSSOStatus
- ImportApplicationInstanceServiceProviderMetadata
- ListApplicationInstances
- ListApplicationInstanceCertificates
- ListApplicationTemplates
- ListDirectoryAssociations
- ListPermissionSets
- ListProfileAssociations
- ListProfiles
• PutMfaDeviceManagementForDirectory
• PutPermissionsPolicy
• StartSSO
• UpdateApplicationInstanceActiveCertificate
• UpdateApplicationInstanceDisplayData
• UpdateApplicationInstanceServiceProviderConfiguration
• UpdateApplicationInstanceStatus
• UpdateApplicationInstanceResponseConfiguration
• UpdateApplicationInstanceResponseSchemaConfiguration
• UpdateApplicationInstanceSecurityConfiguration
• UpdateDirectoryAssociation
• UpdateProfile

The following AWS SSO identity store CloudTrail actions are supported:

• AddMemberToGroup
• CompleteVirtualMfaDeviceRegistration
• CreateAlias
• CreateExternalIdPConfigurationForDirectory
• CreateGroup
• CreateUser
• DeleteExternalIdPConfigurationForDirectory
• DeleteGroup
• DeleteMfaDeviceForUser
• DeleteUser
• DescribeDirectory
• DescribeGroups
• DescribeUsers
• DisableExternalIdPConfigurationForDirectory
• DisableUser
• EnableExternalIdPConfigurationForDirectory
• EnableUser
• GetAWSSPConfigurationForDirectory
• ListExternalIdPConfigurationsForDirectory
• ListGroupsForUser
• ListMembersInGroup
• ListMfaDevicesForUser
• RemoveMemberFromGroup
• SearchGroups
• SearchUsers
• StartVirtualMfaDeviceRegistration
• UpdateExternalIdPConfigurationForDirectory
• UpdateGroup
• UpdatePassword
• UpdateUser
• VerifyEmail

Every log entry contains information about who generated the request. The identity information in the log helps you determine whether the request was made by an AWS account root user or with IAM user credentials. You can also learn whether the request was made with temporary security credentials for a role or federated user or by another AWS service. For more information, see the CloudTrail userIdentity Element.

You can create a trail and store your log files in your Amazon S3 bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted with Amazon S3 server-side encryption (SSE).

To be notified of log file delivery, configure CloudTrail to publish Amazon SNS notifications when new log files are delivered. For more information, see Configuring Amazon SNS Notifications for CloudTrail.

You can also aggregate AWS SSO log files from multiple AWS Regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts.

Understanding AWS SSO Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry for an administrator (samadams@example.com) that took place in the AWS SSO console:

```json
{
"Records": [
{
"eventVersion":"1.05",
"userIdentity": {
  "type":"IAMUser",
  "principalId":"AIDAJAENLMexample",
  "arn":"arn:aws:iam::08966example:user/samadams",
  "accountId":"08966example",
  "accessKeyId":"AKIAIIJM2K4example",
  "userName":"samadams"
},
"eventTime":"2017-11-29T22:39:43Z",
"eventSource":"sso.amazonaws.com",
"eventName":"DescribePermissionsPolicies",
"awsRegion":"us-east-1",
"sourceIPAddress":"203.0.113.0",
"userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36",
"requestParameters": {
  "permissionSetId":"ps-79a0d6e7b95ed05"
},
"responseElements":null,
"requestID":"319ac641-d556-11e7-a34f-69a333106015",
"eventID":"a93952b-13dd-4ae5-a156-d3ad6220b071",
"readOnly":true,
"resources": [

],
"eventType":"AwsApiCall",
"recipientAccountId":"08966example"}
```
The following example shows a CloudTrail log entry for an end-user (bobsmith@example.com) action that took place in the AWS SSO user portal:

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "Unknown",
        "principalId": "example.com//S-1-5-21-1122334455-3652759393-4233131409-1126",
        "accountId": "08966example",
        "userName": "bobsmith@example.com"
      },
      "eventTime": "2017-11-29T18:48:28Z",
      "eventSource": "sso.amazonaws.com",
      "eventName": "https://portal.sso.us-east-1.amazonaws.com/instance/appinstances",
      "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-bc5ed31ce04",
      "eventID": "e6e1f3df-9528-4c6d-a877-6b2b95df91",
      "eventType": "AwsApiCall",
      "recipientAccountId": "08966example"
    }
  ]
}
```

**Amazon CloudWatch Events**

AWS SSO can work with CloudWatch Events to raise events when administrator-specified actions occur in an organization. For example, because of the sensitivity of such actions, most administrators would want to be warned every time someone creates a new account in the organization or when an administrator of a member account attempts to leave the organization. You can configure CloudWatch Events rules that look for these actions and then send the generated events to administrator-defined targets. Targets can be an Amazon SNS topic that emails or text messages its subscribers. You could also create an AWS Lambda function that logs the details of the action for your later review.

To learn more about CloudWatch Events, including how to configure and enable it, see the Amazon CloudWatch Events User Guide.

**Compliance Validation for AWS Single Sign-On**

Third-party auditors assess the security and compliance of AWS Single Sign-On as part of multiple AWS compliance programs.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 SSO is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Config** – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

### Resilience in AWS Single Sign-On

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](https://aws.amazon.com/global-infrastructure/).

### Infrastructure Security in AWS Single Sign-On

As a managed service, AWS Single Sign-On is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](https://aws.amazon.com/security/overview/) whitepaper.

You use AWS published API calls to access AWS SSO through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](https://docs.aws.amazon.com/iot/latest/developerguide/security-webservices.html) (AWS STS) to generate temporary security credentials to sign requests.
Integrating AWS CLI with AWS SSO

AWS Command Line Interface (CLI) 2.0 preview integration with AWS Single Sign-On (AWS SSO) enables developers to sign in directly to the CLI using the same Active Directory or AWS SSO credentials that they normally use to sign in to AWS SSO, and access their assigned accounts and roles. For example, an administrator configures AWS SSO to use Active Directory for authentication, a developer can then sign into the CLI directly using their Active Directory credentials.

AWS CLI integration with AWS SSO offers the following benefits:

- Enterprises can enable their developers to sign in using credentials from AWS SSO or Active Directory by connecting AWS SSO to their Active Directory using AWS Directory Service.
- Developers can sign-in from the CLI for faster access.
- Developers can list and switch between accounts and roles to which they have assigned access.
- Developers can generate and save named role profiles in their CLI configuration automatically and reference them in the CLI to run commands in desired accounts and roles.
- The CLI manages short-term credentials automatically so developers can start in and stay in the CLI securely without interruption, and execute long running scripts.

How to integrate AWS CLI with AWS SSO

To use the AWS CLI integration with AWS SSO, you will need to download the AWS CLI preview version 2.0 of the CLI. For detailed steps on how to download and integrate the CLI with AWS SSO, see Configuring the AWS CLI to use AWS Single Sign-On (AWS SSO) in the AWS Command Line Interface User Guide.
AWS Single Sign-On User Guide

AWS SSO Region Availability

AWS SSO is available in several commonly used AWS Regions. This availability makes it easier for you to configure user access to multiple AWS accounts and business applications. When your users sign in to the user portal, they can select the AWS account that they have permission to. Then they can access the AWS Management Console. For a full list of the Regions that AWS SSO supports, see AWS Regions and Endpoints.

AWS SSO Region Data

When you first enable AWS SSO, all the data that you configure in AWS SSO is stored in the Region where you configured it. This data includes directory configurations, permission sets, application instances, and user assignments to AWS account applications. If you are using the AWS SSO identity store, all users and groups that you create in AWS SSO are also stored in the same Region.

AWS Organizations only supports one AWS SSO Region at a time. If you want to make AWS SSO available in a different Region, you must first delete your current AWS SSO configuration. Switching to a different Region also changes the URL for the user portal.

Delete Your AWS SSO Configuration

When an AWS SSO configuration is deleted, all the data in that configuration is deleted and cannot be recovered. The following table describes what data is deleted based on the directory type that you have currently configured in AWS SSO.

<table>
<thead>
<tr>
<th>What Data Gets Deleted</th>
<th>Connected Directory (AWS Managed Microsoft AD or AD Connector)</th>
<th>AWS SSO Identity Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>All permission sets you have configured for AWS accounts</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>All applications you have configured in AWS SSO</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>All user assignments you have configured for AWS accounts and applications</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>All users and groups in the directory or store</td>
<td>N/A</td>
<td>X</td>
</tr>
</tbody>
</table>

Use the following procedure when you need to delete your current AWS SSO configuration.

**To delete your AWS SSO configuration**

1. Open the AWS SSO console.
2. In the left navigation pane, choose Settings.
3. On the Settings page, under Delete AWS SSO configuration, choose Delete AWS SSO.
4. On the Delete AWS SSO configuration page, select each of the check boxes to acknowledge you understand the data that will be deleted. Type DELETE in the text box, and then choose Delete AWS SSO.
Limits in AWS SSO

The following tables describe limits within AWS SSO.

Application Limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>File size of service provider SAML certificates (in PEM format)</td>
<td>2 kb</td>
</tr>
</tbody>
</table>

AWS Account Limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of permission sets in AWS SSO</td>
<td>500</td>
</tr>
<tr>
<td>Number of permission sets allowed per AWS account</td>
<td>50</td>
</tr>
<tr>
<td>Number of references to AWS managed policies per permission set</td>
<td>10</td>
</tr>
<tr>
<td>Number of inline policies per permission set</td>
<td>1</td>
</tr>
<tr>
<td>Maximum size of inline policy per permission set</td>
<td>10,000 bytes</td>
</tr>
<tr>
<td>Number of IAM roles in the AWS account that can be repaired at a time *</td>
<td>1</td>
</tr>
<tr>
<td>Number of directories that you can have at a time</td>
<td>1</td>
</tr>
</tbody>
</table>

* Permission sets are provisioned in an AWS account as IAM roles. For more information, see Permission Sets (p. 8).

Connected Directory Limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unique Active Directory groups that can be assigned *</td>
<td>1500</td>
</tr>
<tr>
<td>Number of connected directories that you can have at a time</td>
<td>1</td>
</tr>
</tbody>
</table>
* Users within their Active Directory can belong to many directory groups. However within AWS SSO, they can have up to 1500 of their Active Directory groups assigned for using applications.

AWS SSO Identity Store Limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of unique groups that can be assigned *</td>
<td>100</td>
</tr>
<tr>
<td>Number of AWS SSO stores that you can have at a time</td>
<td>1</td>
</tr>
<tr>
<td>Maximum number of users supported in AWS SSO</td>
<td>50000</td>
</tr>
<tr>
<td>Maximum number of groups supported in AWS SSO</td>
<td>10000</td>
</tr>
</tbody>
</table>

* Users within an AWS SSO store can have up to 100 of their groups assigned for using applications.

Additional Limits

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of AWS accounts or applications that can be configured *</td>
<td>500</td>
</tr>
<tr>
<td>Maximum number of unique groups that can be used to evaluate the permissions for a user **</td>
<td>500</td>
</tr>
</tbody>
</table>

* Only 500 AWS accounts or applications (total combined) are supported. For example, you might configure 275 accounts and 225 applications, resulting in a total of 500 accounts and applications.

** Before displaying the user's available AWS accounts and application icons in the user portal, AWS SSO evaluates the user's effective permissions by evaluating their group memberships. Only 500 unique groups can be used to determine a user's effective permissions.
Troubleshooting AWS SSO Issues

The following can help you troubleshoot some common issues you might encounter while setting up or using the AWS SSO console.

Users can’t sign in when their user name is in UPN format

Users might not be able to sign in to the user 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 AWS SSO 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 Enable Multi-Factor Authentication (p. 47). For general information about user name formats used to sign in to Active Directory, see User Name Formats on the Microsoft documentation website.

I cannot get my cloud application configured correctly

Each service provider of a preintegrated cloud application in AWS SSO has its own detailed instruction manual. You can access the manual from the Configuration tab for that application in the AWS SSO console.

If the problem is related to setting up the trust between the service provider's application and AWS SSO, make sure to check the instruction manual for troubleshooting steps.

I don't know what data is in my SAML assertion that would be passed to the service provider

Use the following steps in the user portal to view what data in the SAML assertion will be sent to the application's service provider for the currently signed-in user. This procedure displays the contents in the browser window before sending it to the provider.

1. While you are signed into the portal, hold the Shift key and then choose the application.
2. Examine the information on the page titled You are now in administrator mode.
3. If the information looks good, you can choose Send to <application> to send the assertion to the service provider and review the outcome of the response.
# Document History

The following table describes the documentation for this release of AWS Single Sign-On.

- **Latest documentation update:** November 26, 2019

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for external identity providers</td>
<td>Changed references from directory to identity source, added content to support external identity providers.</td>
<td>November 26, 2019</td>
</tr>
<tr>
<td>New MFA settings</td>
<td>Removed two-step verification topic and added new MFA topic in its place.</td>
<td>October 24, 2019</td>
</tr>
<tr>
<td>New setting to add two-step verification</td>
<td>Added content on how to enable two-step verification for users.</td>
<td>January 16, 2019</td>
</tr>
<tr>
<td>Support for session duration on AWS accounts</td>
<td>Added content on how to set the session duration for an AWS account.</td>
<td>October 30, 2018</td>
</tr>
<tr>
<td>New option to use AWS SSO directory</td>
<td>Added content for choosing either AWS SSO directory or connecting to an existing AD directory.</td>
<td>October 17, 2018</td>
</tr>
<tr>
<td>Support for relay state and session duration on applications</td>
<td>Added content about relay state and session duration for cloud applications.</td>
<td>October 10, 2018</td>
</tr>
<tr>
<td>Additional support for new cloud applications</td>
<td>Added 4me, BambooHR, Bonusly, Citrix ShareFile, ClickTime, Convo, Deputy, Deskpro, Dome9, DruvalnSync, Egnyte, Engagedly, Expensify, Freshdesk, IdeaScale, Igloo, Jitbit, Kudos, LiquidFiles, Lucidchart, PurelyHR, Samanage, ScreenSteps, Sli.do, SmartSheet, Syncplicity, TalentLMS, Trello, UserVoice, Zoho, OpsGenie, DigiCert, WeekDone, ProdPad, and UserEcho to the application catalog.</td>
<td>August 3, 2018</td>
</tr>
<tr>
<td>Support for SSO access to master accounts</td>
<td>Added content about how to delegate SSO access to users in a master account.</td>
<td>July 9, 2018</td>
</tr>
<tr>
<td>Support for new cloud applications</td>
<td>Added DocuSign, Keeper Security, and SugarCRM to the application catalog.</td>
<td>March 16, 2018</td>
</tr>
<tr>
<td>Get temporary credentials for CLI access</td>
<td>Added information about how to get temporary credentials to run AWS CLI commands.</td>
<td>February 22, 2018</td>
</tr>
<tr>
<td>New guide</td>
<td>This is the first release of the AWS SSO User Guide.</td>
<td>December 7, 2017</td>
</tr>
</tbody>
</table>
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.