

Azure AD SSO integration with AWS Single-Account Access

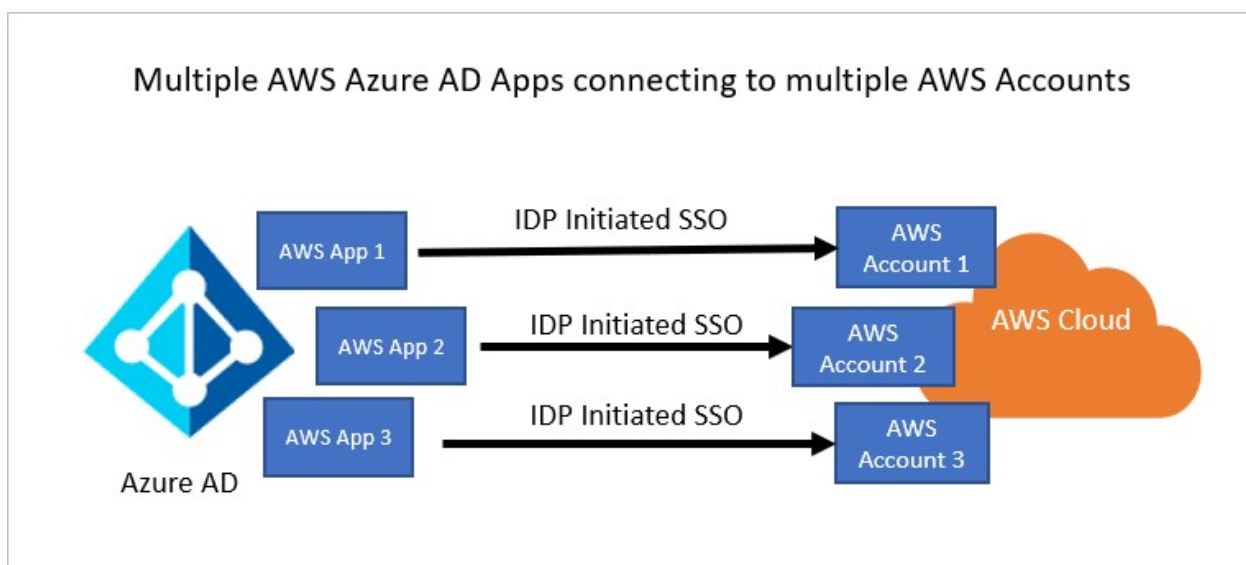
AWS Single Sign-On

[AWS Single Sign-On](#) makes it easy to manage access centrally to multiple AWS accounts and AWS applications, with sign-in through Microsoft Azure AD. Federate Microsoft Azure AD with AWS SSO once, and use AWS SSO to manage permissions across all of your AWS accounts from one place. AWS SSO provisions permissions automatically and keeps them current as you update policies and access assignments. End users can authenticate with their Azure AD credentials to access the AWS Console, Command Line Interface, and AWS SSO integrated applications.

AWS Single-Account Access

[AWS Single-Account Access](#) has been used by customers over the past several years and enables you to federate Azure AD to a single AWS account and use Azure AD to manage access to AWS IAM roles. AWS IAM administrators define roles and policies in each AWS account. For each AWS account, Azure AD administrators federate to AWS IAM, assign users or groups to the account, and configure Azure AD to send assertions that authorize role access.

AWS Single-Account Access architecture



You can configure multiple identifiers for multiple instances. For example:

- <https://signin.aws.amazon.com/saml#1>
- <https://signin.aws.amazon.com/saml#2>

Prerequisites

To get started, you need the following items:

- An Azure AD subscription. If you don't have a subscription, you can get a [free account](#).
- An AWS IAM IdP enabled subscription.
- Along with Cloud Application Administrator, Application Administrator can also add or manage applications in Azure AD. For more information, see [Azure built-in roles](#).

Adding AWS Single-Account Access from the gallery

To configure the integration of AWS Single-Account Access into Azure AD, you need to add AWS Single-Account Access from the gallery to your list of managed SaaS apps.

1. Sign in to the Azure portal using a work account, school account, or personal Microsoft account.
2. In the Azure portal, search for and select Azure Active Directory.
3. Within the Azure Active Directory overview menu, choose Enterprise Applications > All applications.
4. Select New application to add an application.
5. In the Add from the gallery section, type AWS Single-Account Access in the search box.
6. Select AWS Single-Account Access from results panel and then add the app. Wait a few seconds while the app is added to your tenant.

Alternatively, you can also use the [Enterprise App Configuration Wizard](#)

Configure Azure AD SSO

Follow these steps to enable Azure AD SSO in the Azure portal.

1. In the Azure portal, on the AWS Single-Account Access application integration page, find the Manage section and select single sign-on.
2. On the Select a single sign-on method page, select SAML.

3. On the Set up single sign-on with SAML page, click the pencil icon for Basic SAML Configuration to edit the settings.

Set up Single Sign-On with SAML

An SSO implementation based on federation protocols improves security, reliability, and end user experiences and is easier to implement. Choose SAML single sign-on whenever possible for existing applications that do not use OpenID Connect or OAuth. [Learn more.](#)

Read the [configuration guide](#) for help integrating <App Name>

1

Basic SAML Configuration

Edit

Identifier (Entity ID)

Reply URL (Assertion Consumer Service URL)

Sign on URL

Relay State (Optional)

Logout Url (Optional)

4. In the Basic SAML Configuration section, update both Identifier (Entity ID) and Reply URL with the same default value: `https://signin.aws.amazon.com/saml`. You must select Save to save the configuration changes.
5. When you are configuring more than one instance, provide an identifier value. From second instance onwards, use the following format, including a # sign to specify a unique SPN value.
`https://signin.aws.amazon.com/saml#2`
6. AWS application expects the SAML assertions in a specific format, which requires you to add custom attribute mappings to your SAML token attributes configuration. The following screenshot shows the list of default attributes.

2

Attributes & Claims

Edit

givenname	user.givenname
surname	user.surname
emailaddress	user.mail
name	user.userprincipalname
Unique User Identifier	user.userprincipalname

7. In addition to above, AWS application expects few more attributes to be passed back in SAML response which are shown below. These attributes are also pre populated but you can review them as per your requirements.

Name	Source attribute	Namespace
RoleSessionName	user.userprincipalname	<code>https://aws.amazon.com/SAML/Attributes</code>
Role	user.assignedroles	<code>https://aws.amazon.com/SAML/Attributes</code>
SessionDuration	"provide a value between 900 seconds (15 minutes) to 43200 seconds (12 hours)"	<code>https://aws.amazon.com/SAML/Attributes</code>

- On the Set up Single Sign-On with SAML page, in the SAML Signing Certificate section, select Download to download the federation metadata XML file, and then save it to your computer.

3

SAML Signing Certificate

Edit

Status	Active
Thumbprint	<Thumbprintvalue>
Expiration	<Expiration>
Notification Email	<Email address>
App Federation Metadata Url	<input type="text" value="<App Federation Metadata Url>"/>
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

Configure AWS Single-Account Access SSO

- In a different browser window, sign-on to your AWS company site as an administrator.
- In AWS home page, search for IAM and click it.

Q IAM

Search results for 'IAM'

Services (9)

Features (19)

Resources **New**

Blogs (1,538)

Documentation (45,349)

Knowledge Articles (14)


Tutorials (2)

Events (13)


Marketplace (460)

Services


See all 9

 **IAM** ☆


Manage access to AWS resources

 **IAM Identity Center (successor to AWS Single Sign-On)** ☆

Manage workforce user access to multiple AWS accounts and cloud applications


 **Resource Access Manager** ☆

Share AWS resources with other accounts or AWS Organizations

 **Serverless Application Repository** ☆

Assemble, deploy, and share serverless applications within teams or publicly

3. Go to Access management -> Identity Providers and click Add provider button.

Identity and Access Management (IAM) 

Dashboard

▼ Access management

User groups

Users


Roles

Policies

Identity providers


Account settings

IAM > Identity providers

 **Have you considered using AWS IAM Identity Center?**
AWS IAM Identity Center [makes it easy to centrally manage access to multiple AWS accounts and provide users with single sign-on access to all their assigned accounts from one place. With IAM Identity Center, you can create and manage user identities in IAM Identity Center or easily connect to your existing SAML 2.0 compatible identity provider.](#)
[Learn more](#)

Identity providers (4) [Info](#)

Use an identity provider (IdP) to manage your user identities outside of AWS, but grant the user identities permissions to use AWS resources in your account.

< 1 > 

Provider	Type	Creation time
----------	------	---------------

4. In the Add an Identity provider page, perform the following steps:

[IAM](#) > [Identity providers](#) > [Create Identity Provider](#)

Add an Identity provider

Configure provider

Provider type [Info](#)

☒ **SAML**
Establish trust between your AWS account and a SAML 2.0 compatible Identity Provider such as Shibboleth or Active Directory Federation Services.

☐ **OpenID Connect**
Establish trust between your AWS account and Identity Provider services, such as Google or Salesforce.

Provider name
Enter a meaningful name to identify this provider

Maximum 128 characters. Use alphanumeric or '._-' characters.

Metadata document [Info](#)
This document is issued by your IdP.

Choose file

File needs to be a valid UTF-8 XML document.

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

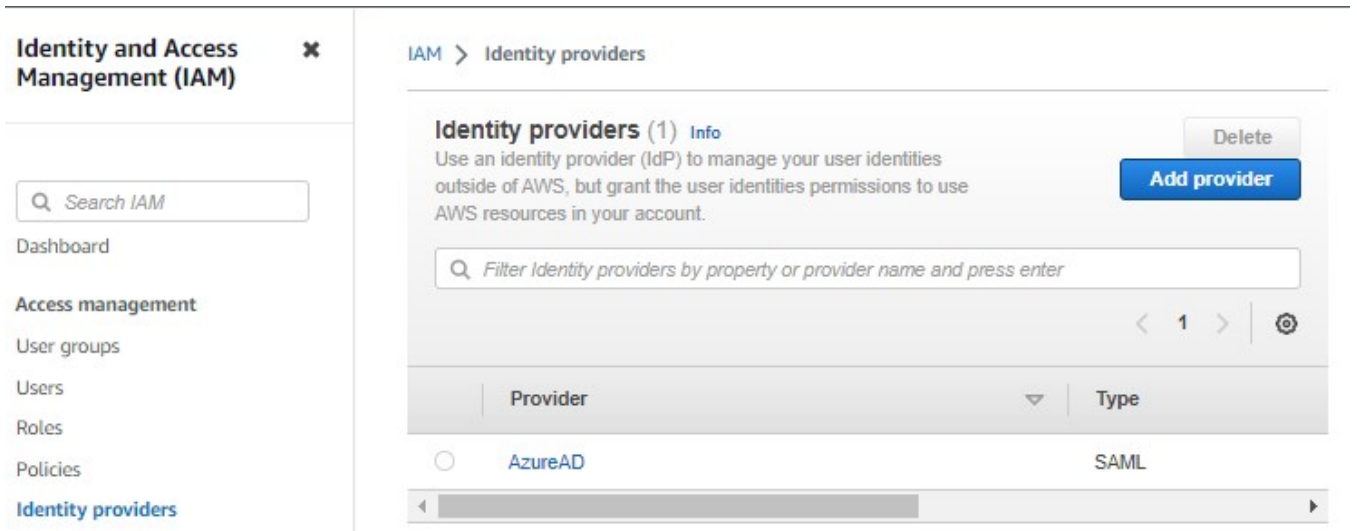
No tags associated with the resource.

Add tag

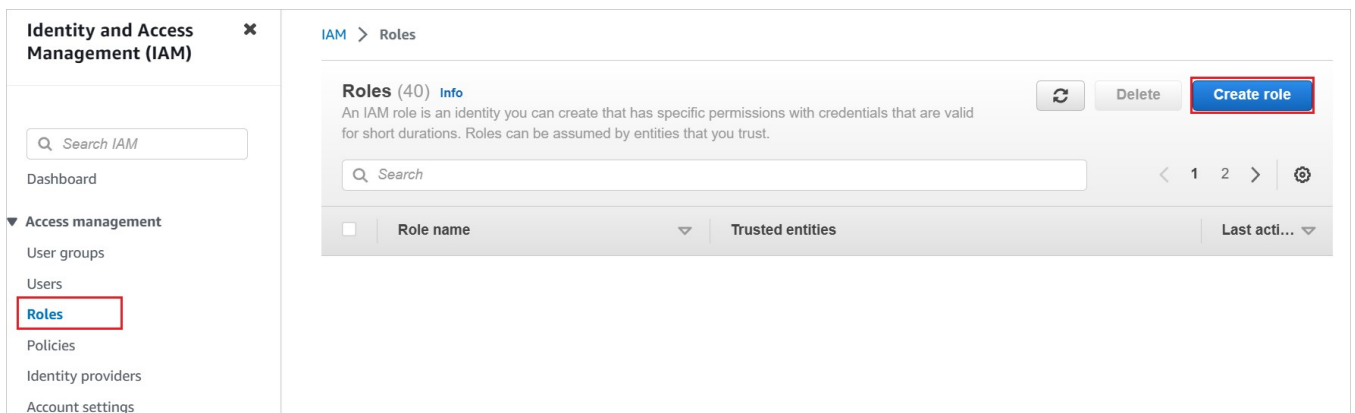
You can add up to 50 more tags.

[Cancel](#) [Add provider](#)

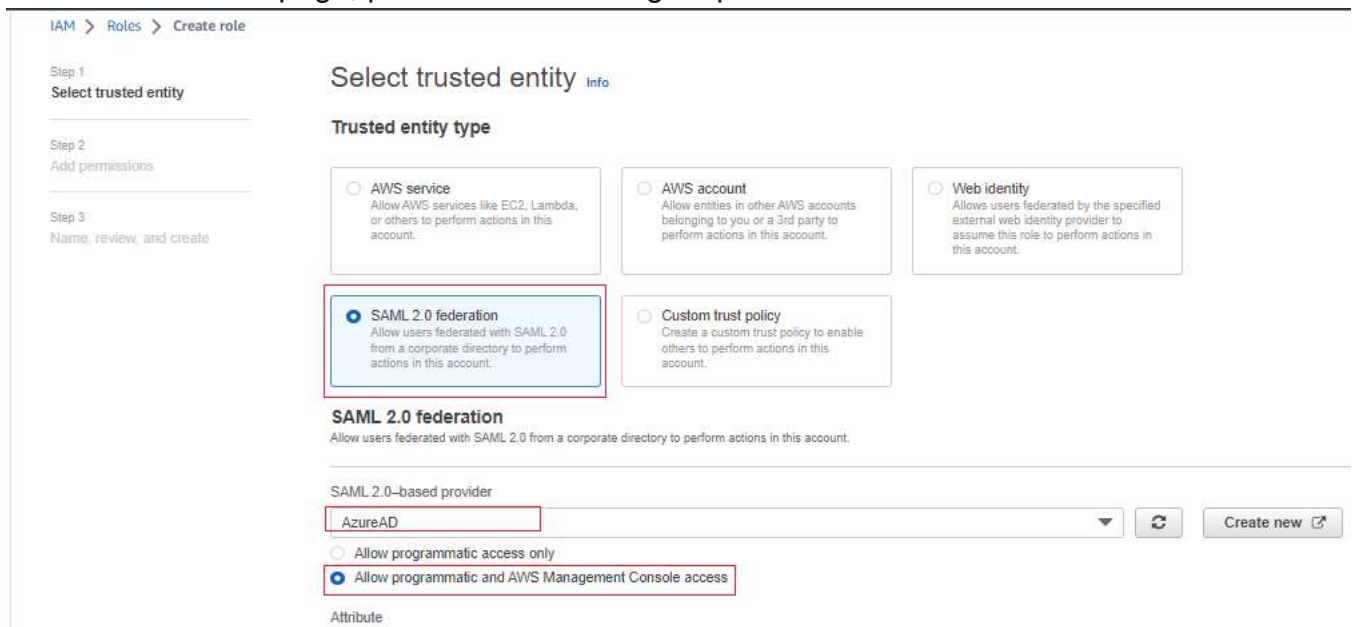
- For Provider type, select SAML.
- For Provider name, type a provider name (for example: *AzureAD*).
- To upload your downloaded metadata file from the Azure portal, select Choose file.
- Click Add provider.



5. Select Roles > Create role.



6. On the Create role page, perform the following steps:



- Choose Trusted entity type, select SAML 2.0 federation.
- Under SAML 2.0 based provider, select the SAML provider you created previously

(for example: *AzureAd*).

c. Select Allow programmatic and AWS Management Console access.

d. Select Next.

7. On the Permissions policies dialog box, attach the appropriate policy, per your organization. Then select Next.

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions [Info](#)

Permissions policies (880) [Info](#)

Choose one or more policies to attach to your new role.

< 1 2 3 4 5 6 7 ... 44 >

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch
<input type="checkbox"/>	AmazonGrafanaAmazo...	Custom...	Allows Amazon Grafana to access Amazon Elasticsearch

► Set permissions boundary - optional [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

8. On the Review dialog box, perform the following steps:

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+', '@', '_' characters.

Description

Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+', '@', '_' characters.

Step 1: Select trusted entities

Edit

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRoleWithSAML",
7       "Principal": {
8         "Federated": "arn:aws:iam::<Account_ID>:saml-provider/WAAD"
9       },
10      "Condition": {
11        "StringEquals": {
12          "SAML:aud": [
13            "https://signin.aws.amazon.com/saml"
14          ]
15        }
16      }
17    }
18  ]
19 }
```

Step 2: Add permissions

Edit

Permissions policy summary

Policy name ↗	Type	Attached as
-------------------------------	------	-------------

Tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel

Previous

Create role

- In Role name, enter your role name example **SysOps-Viewer** .
- In Description, enter the role description.
- Select Create role.
- Create as many roles as needed and map them to the Azure identity provider.

<input type="checkbox"/>	Role name	Trusted entities
<input type="checkbox"/>	billing-readonly	Identity Provider: am:aws:iam::012094719592:saml-provider/AzureAD
<input type="checkbox"/>	SysOps-Role	Identity Provider: am:aws:iam::012094719592:saml-provider/AzureAD
<input type="checkbox"/>	SysOps-Viewer	Identity Provider: am:aws:iam::012094719592:saml-provider/AzureAD

Note: later these roles will be added to Azure Provisioning.

9. Use AWS service account credentials for fetching the roles from the AWS account in Azure AD user provisioning. For this, open the AWS console home.
10. In the IAM section, select Policies and click Create policy.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings

Policies (1119) Info

A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press enter.

Policy name	Type	Used as	Description
AmazonGrafanaAmazonOpenSearchPolicy-78xBldQvH	Customer managed	Permissions policy (1)	Allows Amazon Grafar
AmazonGrafanaAmazonOpenSearchPolicy-7XKZU2fUz	Customer managed	Permissions policy (1)	Allows Amazon Grafar
AmazonGrafanaAmazonOpenSearchPolicy-81FyA6lcx	Customer managed	Permissions policy (1)	Allows Amazon Grafar
AmazonGrafanaAmazonOpenSearchPolicy-9w4plFNRT	Customer managed	Permissions policy (1)	Allows Amazon Grafar

Create policy

11. Create your own policy to fetch all the roles from AWS accounts.

Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor **JSON** Import managed policy

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "iam:ListRoles"
8       ],
9       "Resource": "*"
10    }
11  ]
12 }

```

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Character count: 99 of 6,144.

Cancel **Next: Tags**

- a. In Create policy, select the JSON tab.
- b. In the policy document, add the following JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles"
      ],
      "Resource": "*"
    }
  ]
}
```

- c. Click **Next: Tags**.

12. You can also add the required tags in the below page and click Next: Review.

Create policy

123

Add tags - optional
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

13. Define the new policy.

Create policy

123

Review policy

Name*

Use alphanumeric and '+=, @_-' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+=, @_-' characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 370 services) Show remaining 369			
IAM	Limited: List	All resources	None

Tags

Key	Value
No tags associated with the resource.	

* Required

[Cancel](#) [Previous](#) [Create policy](#)

- For Name, enter AzureAD_SSOUserRole_Policy.
- For Description, enter This policy will allow to fetch the roles from AWS accounts.
- Select Create policy.

14. Create a new user account in the AWS IAM service.

- In the AWS IAM console, select Users and click Add users.

Identity and Access Management (IAM)

[Dashboard](#)
[Access management](#)
[User groups](#)
[Users](#)
[Roles](#)
[Policies](#)
[Identity providers](#)
[Account settings](#)

IAM > Users

Ready to streamline human access to AWS and cloud apps?

[Dismiss](#) [Manage workforce users](#)

Identity Center is enabled. We recommend managing workforce users' access to AWS accounts and cloud applications in Identity Center.

[Learn more](#) | [Watch how it works](#)

Users (5) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

[Refresh](#) [Delete](#) [Add users](#)

- In the Specify user details section, enter the user name as AzureADRoleManager

and select Next.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

i If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Notes: Do not select "Aws Managment Console Access" because this credential only using by Azure ADProvsioning.

c. Create a new policy for this user.

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1121)

Choose one or more policies to attach to your new user.

Filter distributions by text, property or value

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerService...	AWS managed	0
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	6
<input type="checkbox"/>	AdministratorAccess-A...	AWS managed	0
<input type="checkbox"/>	AdministratorAccess-A...	AWS managed	0

► **Permissions boundary - optional**
Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

Cancel Previous **Next**

- Select Attach existing policies directly.
- Search for the newly created policy in the filter section AzureAD_SSOUserRole_Policy.
- Select the policy, and then select Next.

15. Review your choices and select Create user.


16. To download the user credentials of a user, enable the console access in Security credentials tab.

[IAM](#) > [Users](#) > [AzureADRoleManager](#)

AzureADRoleManager

Delete

Summary

ARN  <code>arn:aws:iam::<Account_ID>:user/AzureADRoleManager</code>	Console access Disabled	Access key 1 Not enabled
Created	Last console sign-in -	Access key 2 Not enabled

Permissions

Groups


Tags

Security credentials

Access Advisor


Console sign-in


Enable console access


Console sign-in link  <code>https://<Account_ID>.signin.aws.amazon.com/console</code>	Console password Not enabled
---	---------------------------------


17. Enter these credentials into the Azure AD user provisioning section to fetch the roles from the AWS console.

Console password ✕

 **You have successfully enabled the user's new password.**
This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

Console sign-in URL
 [https:// <Account_ID> .signin.aws.amazon.com/console](https://<Account_ID>.signin.aws.amazon.com/console)

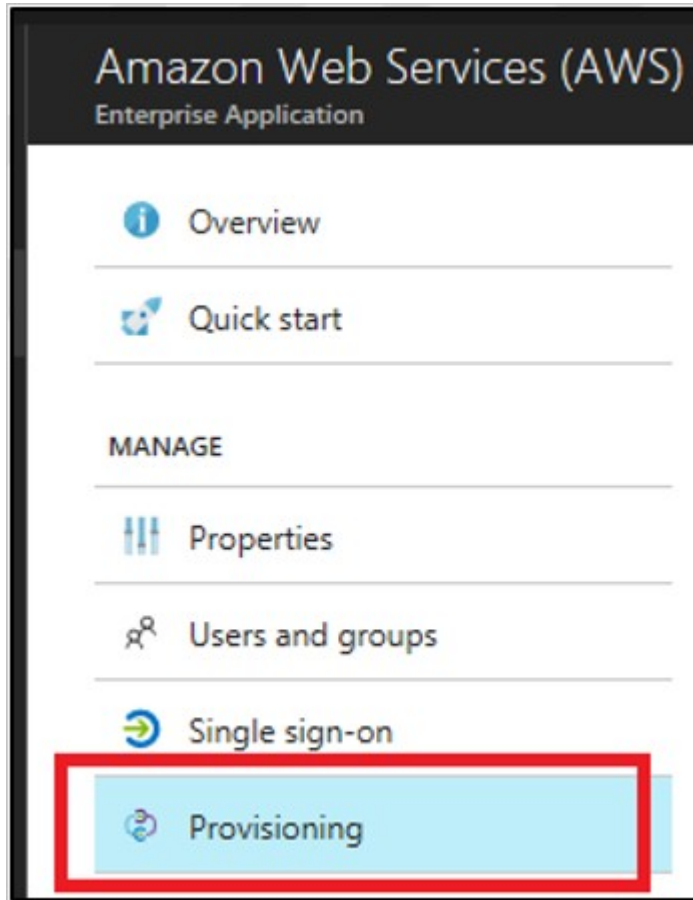
User name
 AzureADRoleManager

Console password
 ***** [Show](#)

[Download .csv file](#) [Close](#)

Configure role provisioning in AWS Single-Account Access

1. In the Azure AD management portal, in the AWS app, go to Provisioning.

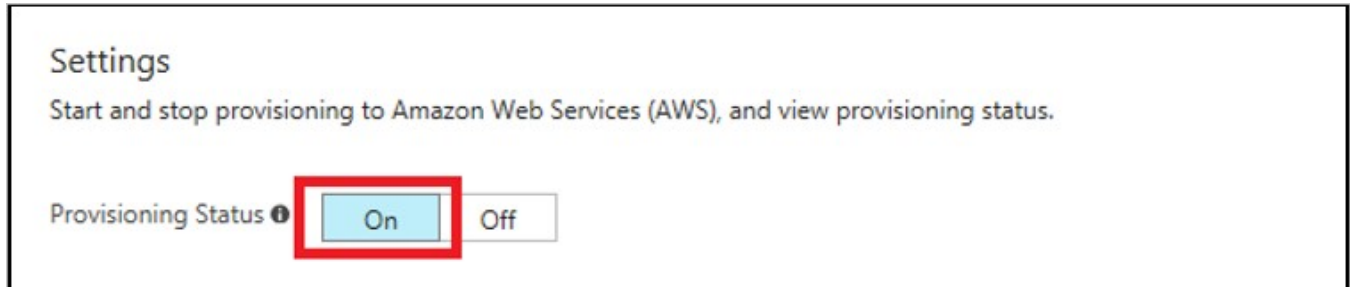


2. Enter the access key and secret in the clientsecret and Secret Token fields, respectively.

A screenshot of the 'Admin Credentials' section in the AWS app configuration. The section title is 'Admin Credentials'. Below it, a message states: 'Azure AD needs the following information to connect to Amazon Web Services (AWS)'s API and synchronize user data.' There are two input fields: the first is labeled '* clientsecret' and the second is labeled '* Secret Token'. Both fields contain masked text represented by dots. To the right of each field is a green checkmark icon. Below the input fields is a button labeled 'Test Connection'. Red rectangular boxes highlight the 'clientsecret' input field, the 'Secret Token' input field, and the 'Test Connection' button.

- a. Enter the AWS user access key in the clientsecret field.
- b. Enter the AWS user secret in the Secret Token field.
- c. Select Test Connection.
- d. Save the setting by selecting Save.

3. In the Settings section, for Provisioning Status, select On. Then select Save.



Create Azure AD user for AWS console access

In this section, you'll create a test user in the Azure portal called Maruf.

1. In the Azure portal, search for and select Azure Active Directory.
2. Within the Azure Active Directory overview menu, choose Users > All users.
3. Select New user at the top of the screen.
4. In the User properties, follow these steps:
 - a. In the Name field, enter `maruf`
 - b. In the User name field, enter the username@companydomain.extension. For example, `maruf@companydomain.com`.
 - c. Select the Show password check box, and then write down the value that's displayed in the Password box.
 - d. Click Create.

Assign the Azure AD user or group

In this section, you'll enable Maruf to use Azure single sign-on by granting access to AWS Single-Account Access.

1. In the Azure portal, select Enterprise Applications, and then select All applications.
2. In the applications list, select AWS Single-Account Access.
3. In the app's overview page, find the Manage section and select Users and groups.
4. Select Add user, then select Users and groups in the Add Assignment dialog.
5. In the Users and groups dialog, select B.Simon from the Users list, then click the Select button at the bottom of the screen.
6. If you are expecting a role to be assigned to the users, you can select it from the Select a role dropdown. If no role has been set up for this app, you see "Default Access" role selected.
7. In the Add Assignment dialog, click the Assign button.

... > iLearn-AWS Single-Account Access | Users and groups

Edit Assignment

Users and groups

1 user selected.

Select a role

None Selected

Assign

Select a role

Only a single role can be selected

Disabled roles cannot be selected.

AzureAD-Provisioning,AzureAD

billing-readonly,AzureAD

SysOps-Role,AzureAD

SysOps-Viewer,AzureAD

Selected Role

Select

iLearn-AWS Single-Account Access | Users and groups

Enterprise Application

Overview

Deployment Plan

Diagnose and solve problems

Manage

Properties

Owners

Roles and administrators

Users and groups

Single sign-on

Provisioning

+ Add user/group | Edit assignment | Remove

The application will appear for assigned users within My Apps. Set 'visible to users?' to no in properties to prevent this.

Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the [application registration](#).

	Display Name	Object Type	Role assigned
<input type="checkbox"/>	AL [User]	User	billing-readonly,AzureAD
<input type="checkbox"/>	SY SySOps-Viewer	Group	SysOps-Viewer,AzureAD
<input type="checkbox"/>	MH Maruf H...	User	SysOps-Role,AzureAD

Test SSO

In this section, you test your Azure AD single sign-on configuration with following options.

After log in to <https://myapplications.microsoft.com/> using you Azure ID you will find the aws application name to log in AWS console.

