Tutorial: Azure Active Directory integration with Amazon Web Services

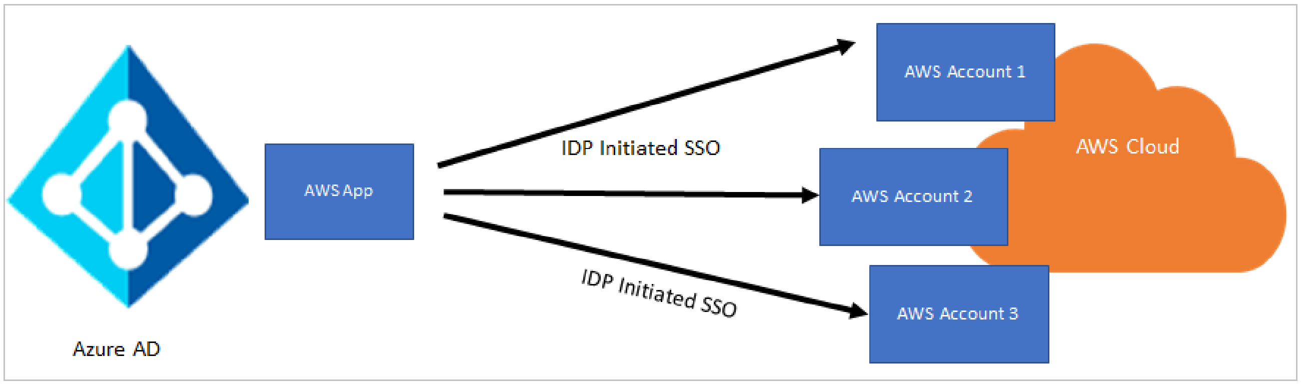
Article • 11/10/2021 • 11 minutes to read • 15 contributors

In this article

In this tutorial, you learn how to integrate Azure Active Directory (Azure AD) with Amazon Web Services (AWS) (legacy tutorial).

This integration provides the following benefits:

- You can control in Azure AD who has access to AWS.
- You can enable your users to automatically sign in to AWS by using single sign-on (SSO) with their Azure AD accounts.
- You can manage your accounts in one central location, the Azure portal.



ⓘ **Note**

We recommend that you *not* connect one AWS app to all your AWS accounts. Instead, we recommend that you use **Azure AD SSO integration with AWS** to configure multiple instances of your AWS account to multiple instances of AWS apps in Azure AD.

We recommend that you *not* connect one AWS app to all your AWS accounts, for the following reasons:

- Use this approach only if you have a small number of AWS accounts and roles, because this model isn't scalable as the number of AWS accounts and the roles within them increase. The approach doesn't use AWS role-import functionality with Azure AD user provisioning, so you have to manually add, update, or delete the roles.

- You have to use the Microsoft Graph Explorer approach to patch all the roles to the app. We don’t recommend using the manifest file approach.

- Customers report that after they've added ~1,200 app roles for a single AWS app, any further operation on the app starts throwing the errors related to size. There is a hard size limit to the application object.

- You have to manually update the roles as they get added in any of the accounts. This is unfortunately a *replace* approach, not an *append* approach. Also, if your account numbers are growing, this becomes an *n* × *n* relationship with accounts and roles.

- All the AWS accounts use the same federation metadata XML file. At the time of certificate rollover, updating the certificate on all the AWS accounts at the same time can be a massive exercise.

Prerequisites

To configure Azure AD integration with AWS, you need the following items:

- An Azure AD subscription. If you don't have an Azure AD subscription, you can get a one-month trial .
- An AWS SSO-enabled subscription.

ⓘ **Note**

We do not recommend that you test the steps in this tutorial in a production environment unless it is necessary.

Scenario description

In this tutorial, you configure and test Azure AD SSO in a test environment.

AWS supports SP-initiated and IDP-initiated SSO.

Add AWS from the gallery

To configure the integration of AWS into Azure AD, you add AWS from the gallery to your list of managed software as a service (SaaS) apps.

1. Sign in to the Azure portal by using either a work or school account, or a personal Microsoft account.

2. On the left pane, select the Azure AD service you want to work with.

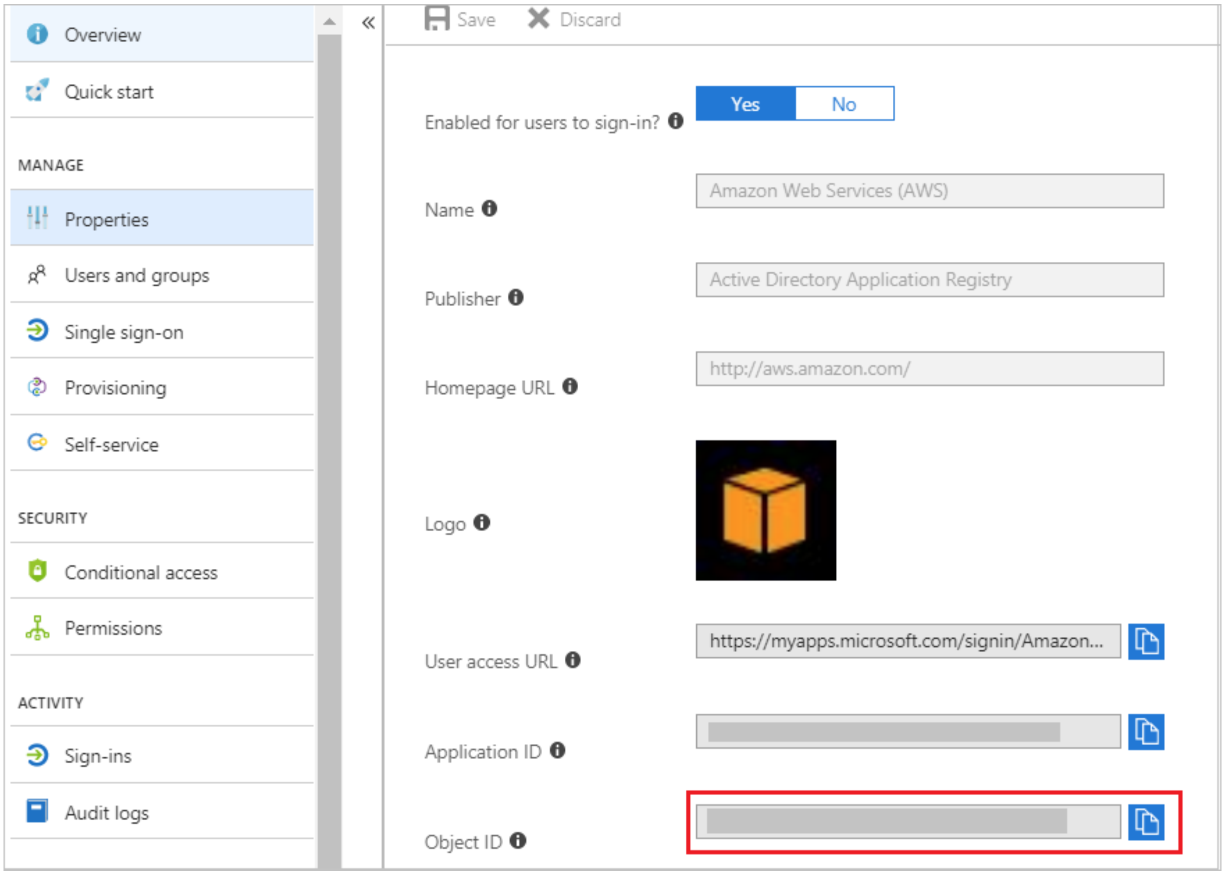
3. Go to **Enterprise Applications**, and then select **All Applications**.

4. To add an application, select **New application**.

5. In the **Add from the gallery** section, type **Amazon Web Services** in the search box.

6. In the results list, select **Amazon Web Services**, and then add the app. In a few seconds, the app is added to your tenant.

7. Go to the **Properties** pane, and then copy the value that's displayed in the **Object ID** box.



Enabled for users to sign-in? Yes No

Name: Amazon Web Services (AWS)

Publisher: Active Directory Application Registry

Homepage URL: http://aws.amazon.com/

Logo

User access URL: https://myapps.microsoft.com/signin/Amazon...

Application ID: [redacted]

Object ID: [redacted]

Configure and test Azure AD SSO

In this section, you configure and test Azure AD single sign-on with AWS based on a test user called "Britta Simon."

For single sign-on to work, Azure AD needs to know what the counterpart user in AWS is to the Azure AD user. In other words, a link relationship between the Azure AD user and the same user in AWS needs to be established.

In AWS, assign the value of the **user name** in Azure AD as the value of the AWS **Username** to establish the link relationship.

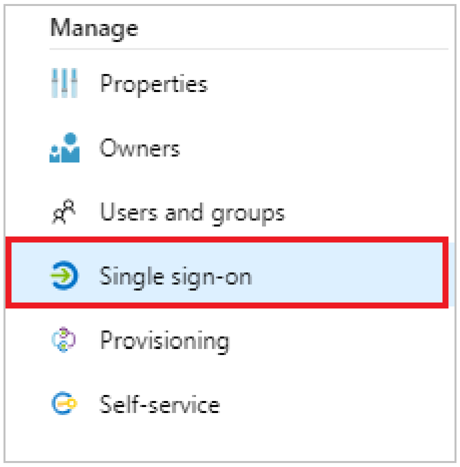
To configure and test Azure AD single sign-on with AWS, do the following:

1. Configure Azure AD SSO to enable your users to use this feature.
2. Configure AWS SSO to configure SSO settings on the application side.
3. Test SSO to verify that the configuration works.

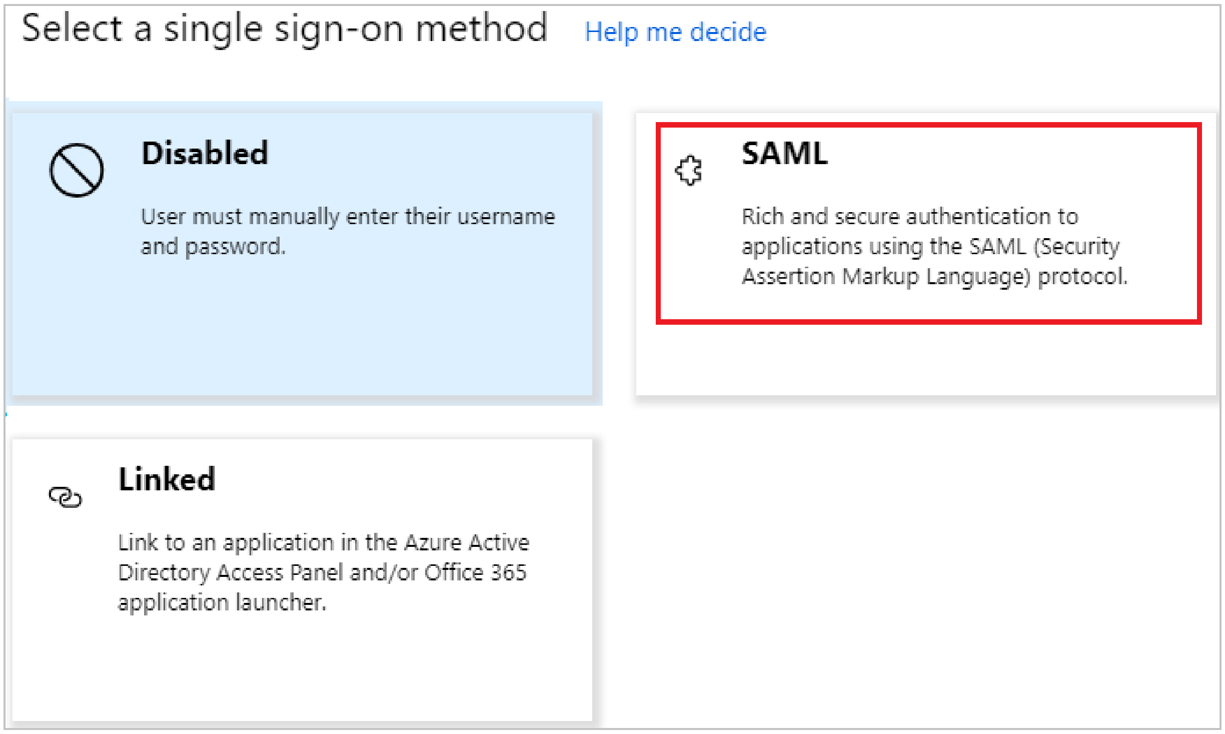
Configure Azure AD SSO

In this section, you enable Azure AD SSO in the Azure portal and configure SSO in your AWS application by doing the following:

1. In the Azure portal, on the left pane of the **Amazon Web Services (AWS)** application integration page, select **Single sign-on**.



2. On the **Select a single sign-on method** pane, select **SAML/WS-Fed** mode to enable single sign-on.

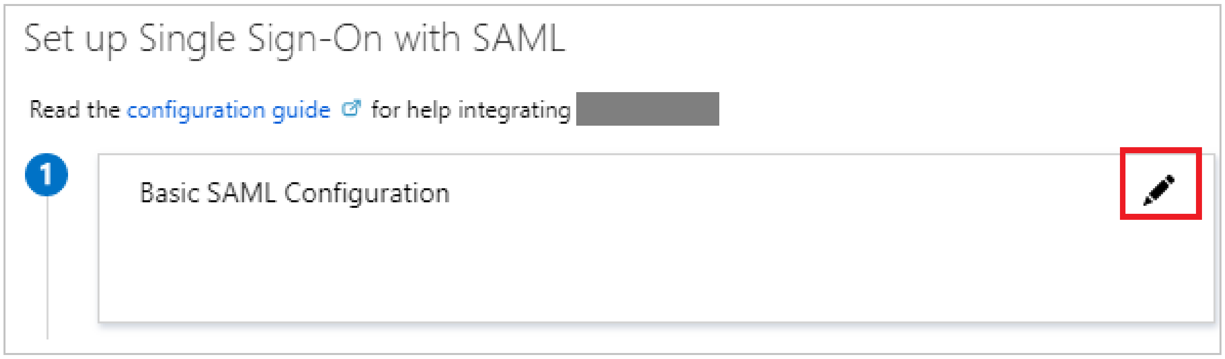

Select a single sign-on method Help me decide

Disabled
User must manually enter their username and password.

SAML
Rich and secure authentication to applications using the SAML (Security Assertion Markup Language) protocol.

Linked
Link to an application in the Azure Active Directory Access Panel and/or Office 365 application launcher.

3. On the **Set up Single Sign-On with SAML** pane, select the **Edit** button (pencil icon).



Set up Single Sign-On with SAML

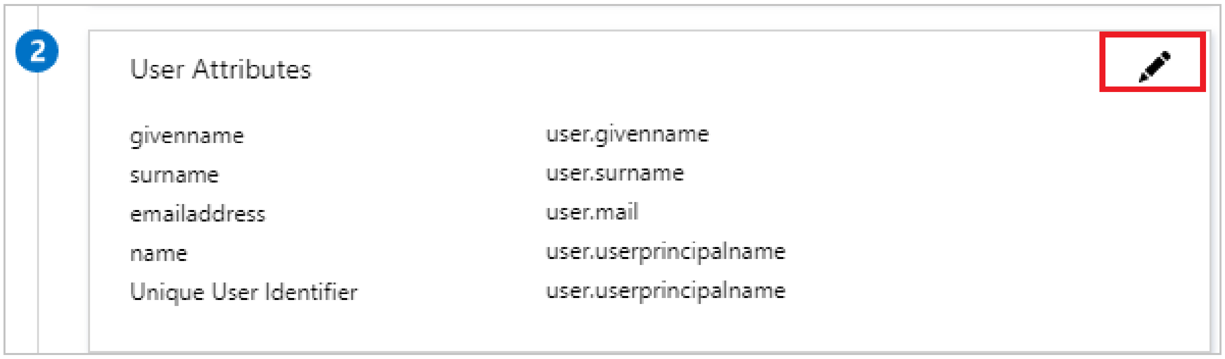
Read the configuration guide for help integrating [redacted]

1 Basic SAML Configuration

4. The **Basic SAML Configuration** pane opens. Skip this section, because the app is preintegrated with Azure. Select **Save**.

   The AWS application expects the SAML assertions in a specific format. You can manage the values of these attributes from the **User Attributes & Claims** section on the **Application integration** page.

5. On the **Set up Single Sign-On with SAML** page, select the **Edit** button.



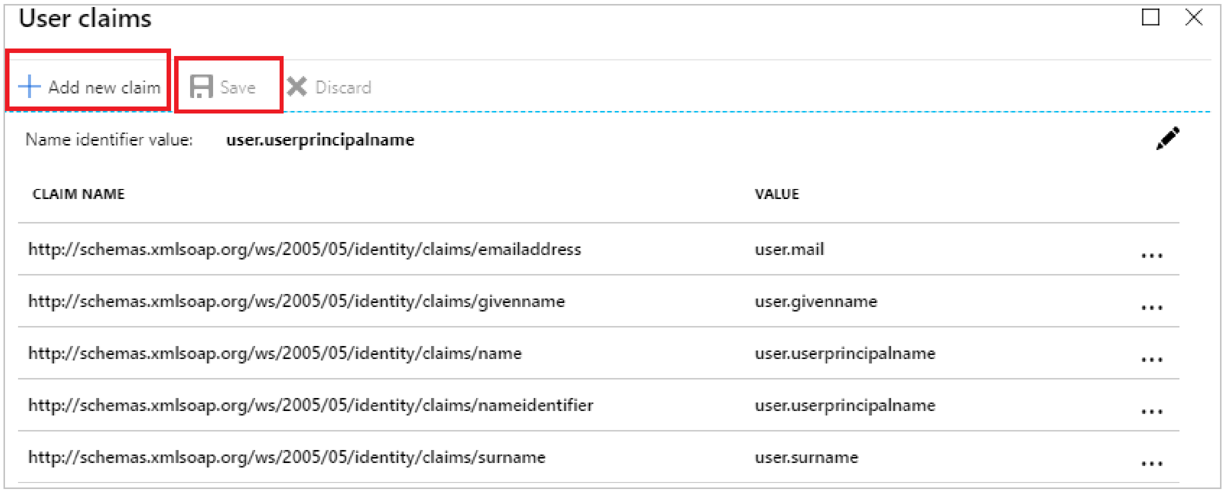
2 User Attributes

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

6. In the **User Claims** section of the **User Attributes** pane, configure the SAML token attribute by using the values in the following table:

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

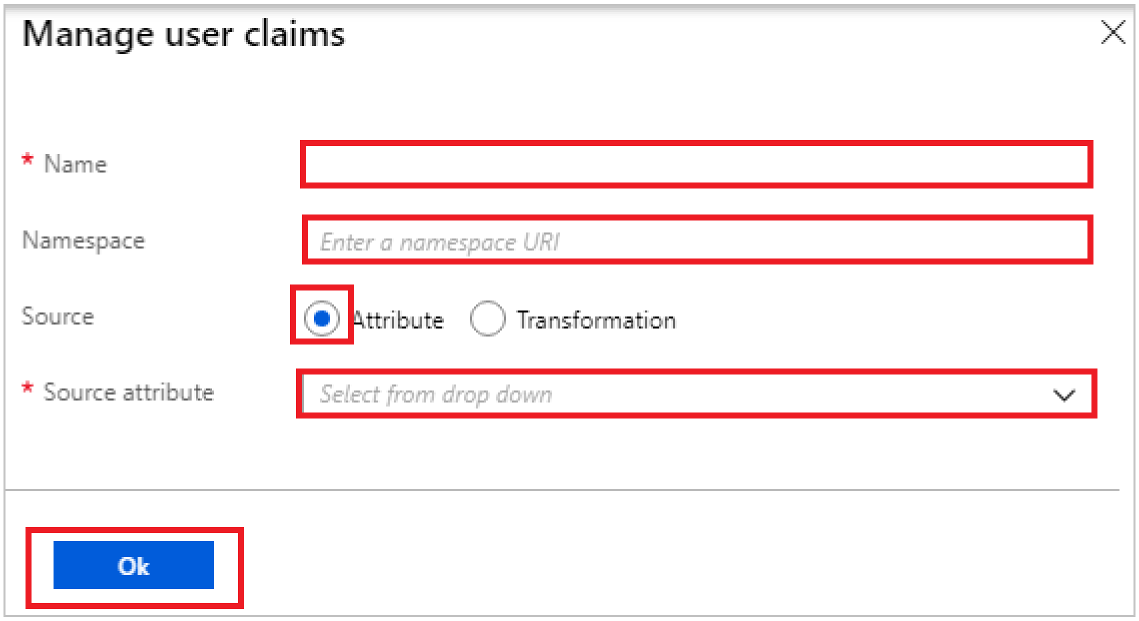
a. Select **Add new claim** and then, on the **Manage user claims** pane, do the following:



User claims

Name identifier value: **user.userprincipalname**

CLAIM NAME	VALUE
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	user.mail
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	user.givenname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	user.userprincipalname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	user.userprincipalname
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	user.surname



Manage user claims

* Name

Namespace: Enter a namespace URI

Source: ● Attribute ○ Transformation

* Source attribute: Select from drop down

b. In the **Name** box, enter the attribute name.

c. In the **Namespace** box, enter the namespace value.

d. For the **Source**, select **Attribute**.

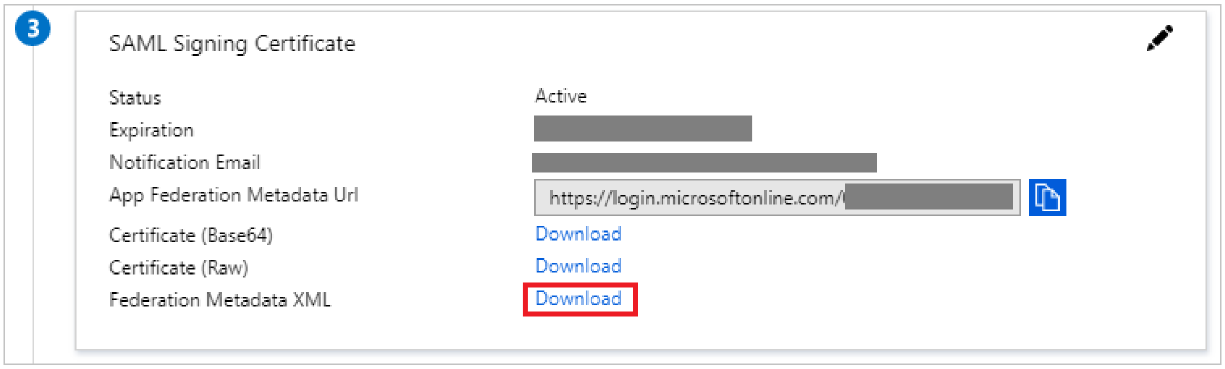
e. In the **Source attribute** drop-down list, select the attribute.

f. Select **Ok**, and then select **Save**.

ⓘ Note

For more information about roles in Azure AD, see Add app roles to your application and receive them in the token.

7. 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.



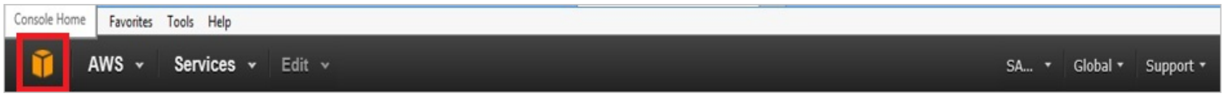
SAML Signing Certificate

Status	Active
Expiration	[redacted]
Notification Email	[redacted]
App Federation Metadata Url	https://login.microsoftonline.com/[redacted]
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

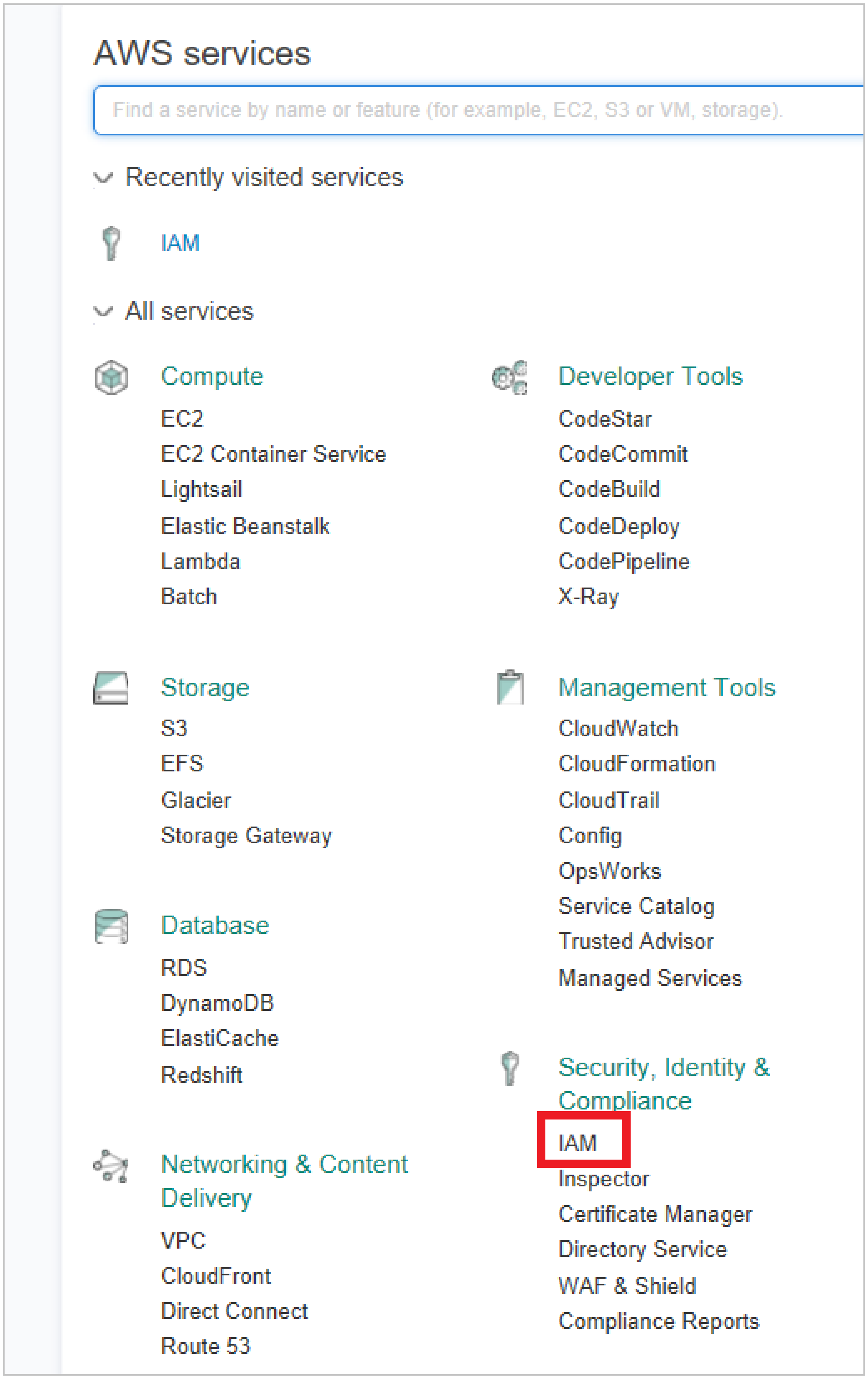
Configure AWS SSO

1. In a new browser window, sign in to your AWS company site as administrator.

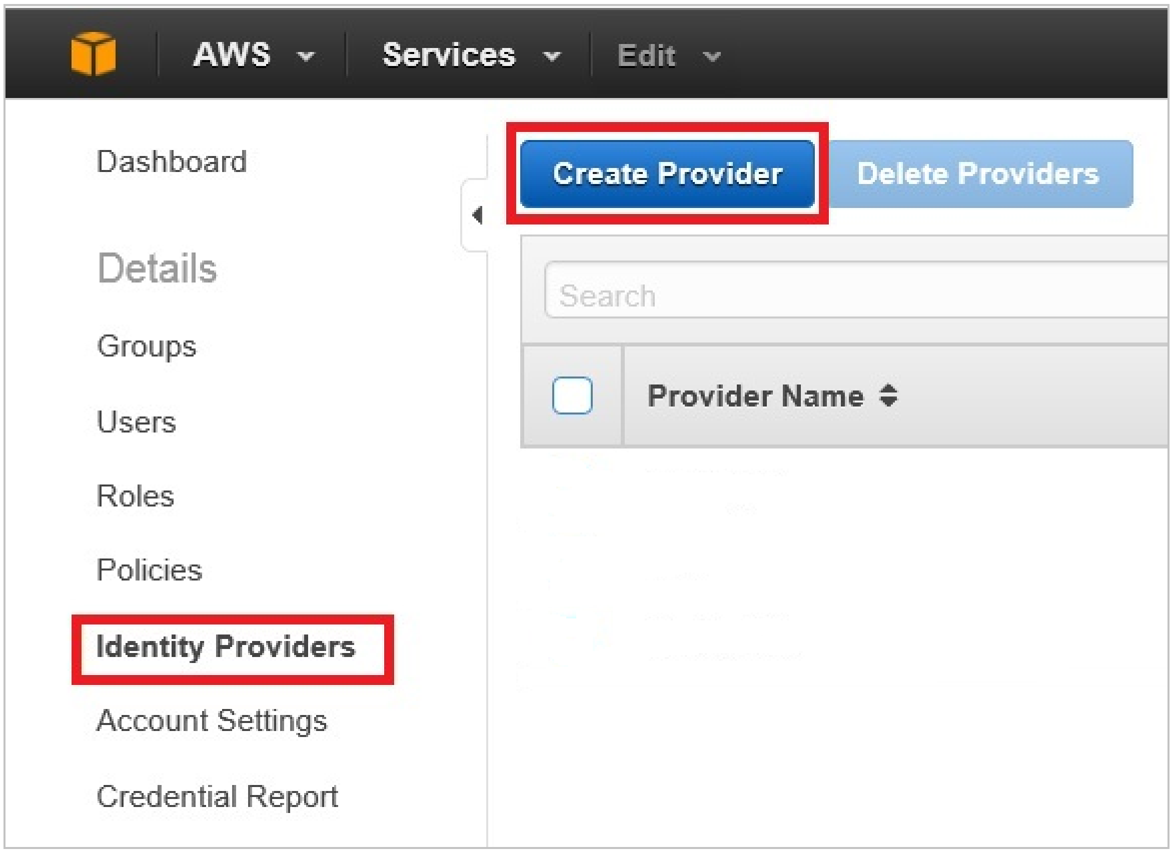
2. Select the **AWS Home** icon.



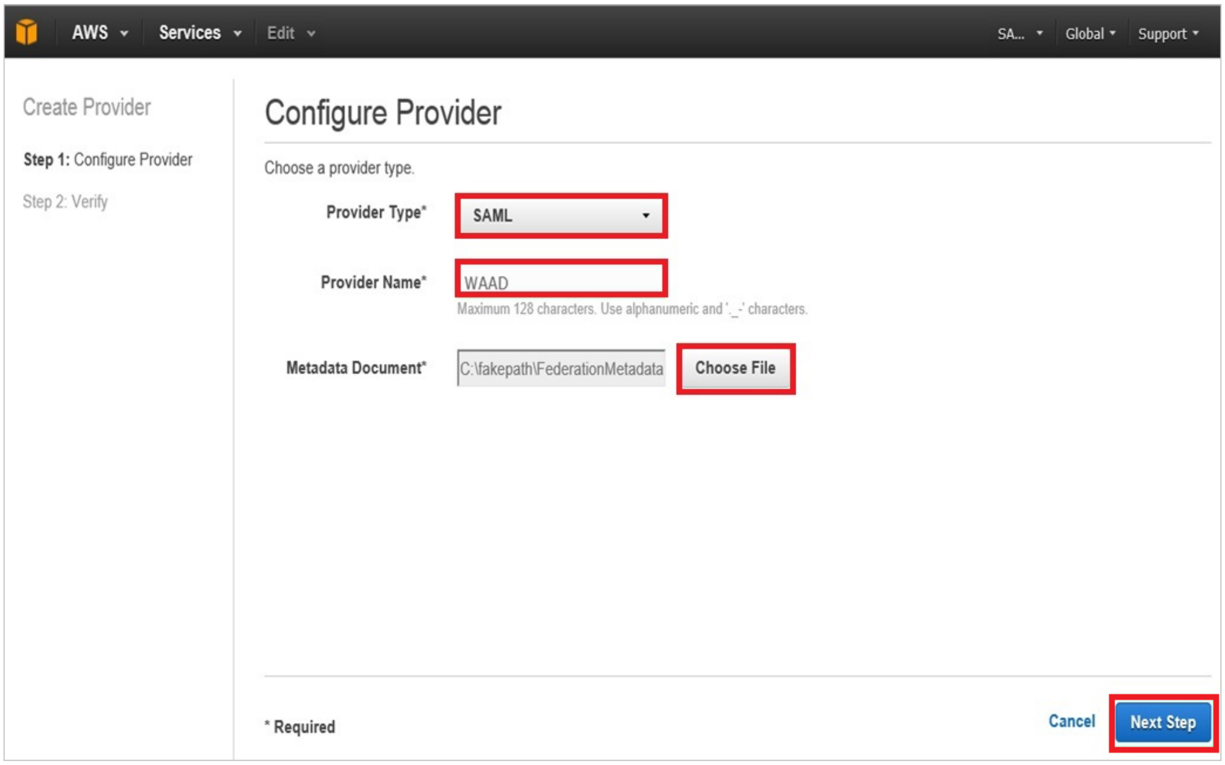
3. On the **AWS services** pane, under **Security, Identity & Compliance**, select **IAM (Identity & Access Management)**.



4. On the left pane, select **Identity Providers**, and then select **Create Provider**.



5. On the **Configure Provider** pane, do the following:



Provider Type*: SAML

Provider Name*: WAAD

Metadata Document*: C:\fakepath\FederationMetadata

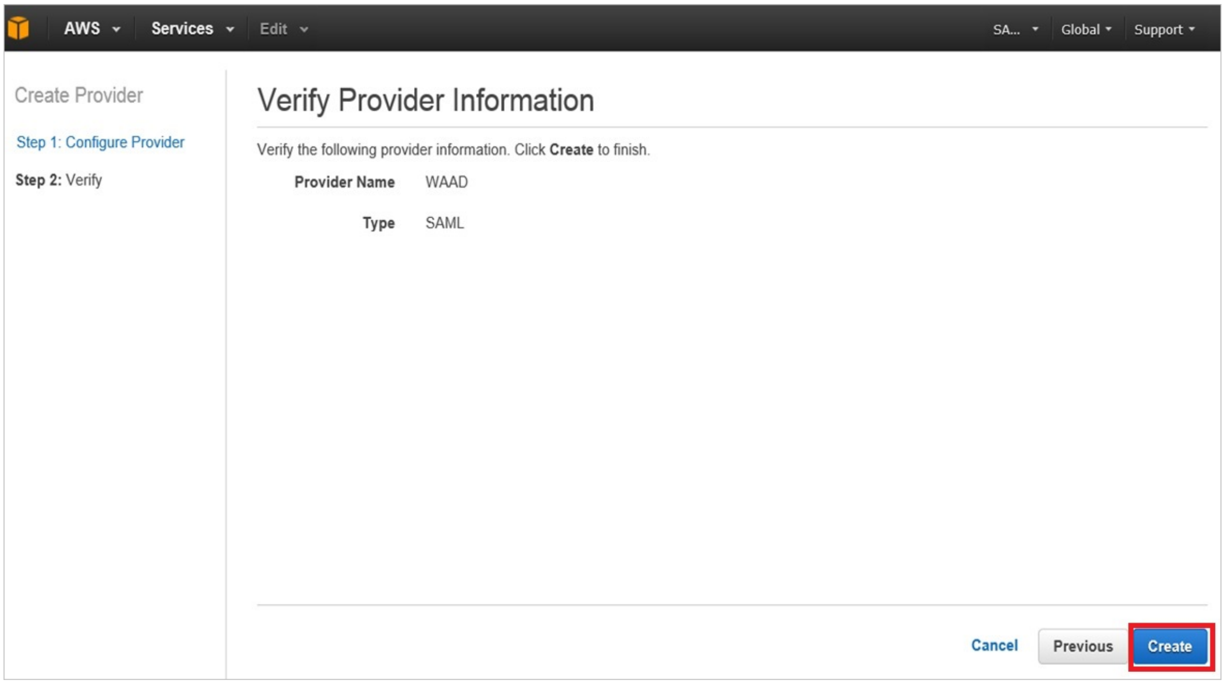
a. In the **Provider Type** drop-down list, select **SAML**.

b. In the **Provider Name** box, enter a provider name (for example. *WAAD*).

c. Next to the **Metadata Document** box, select **Choose File** to upload your downloaded federation metadata XML file to the Azure portal.

d. Select **Next Step**.

6. On the **Verify Provider Information** pane, select **Create**.



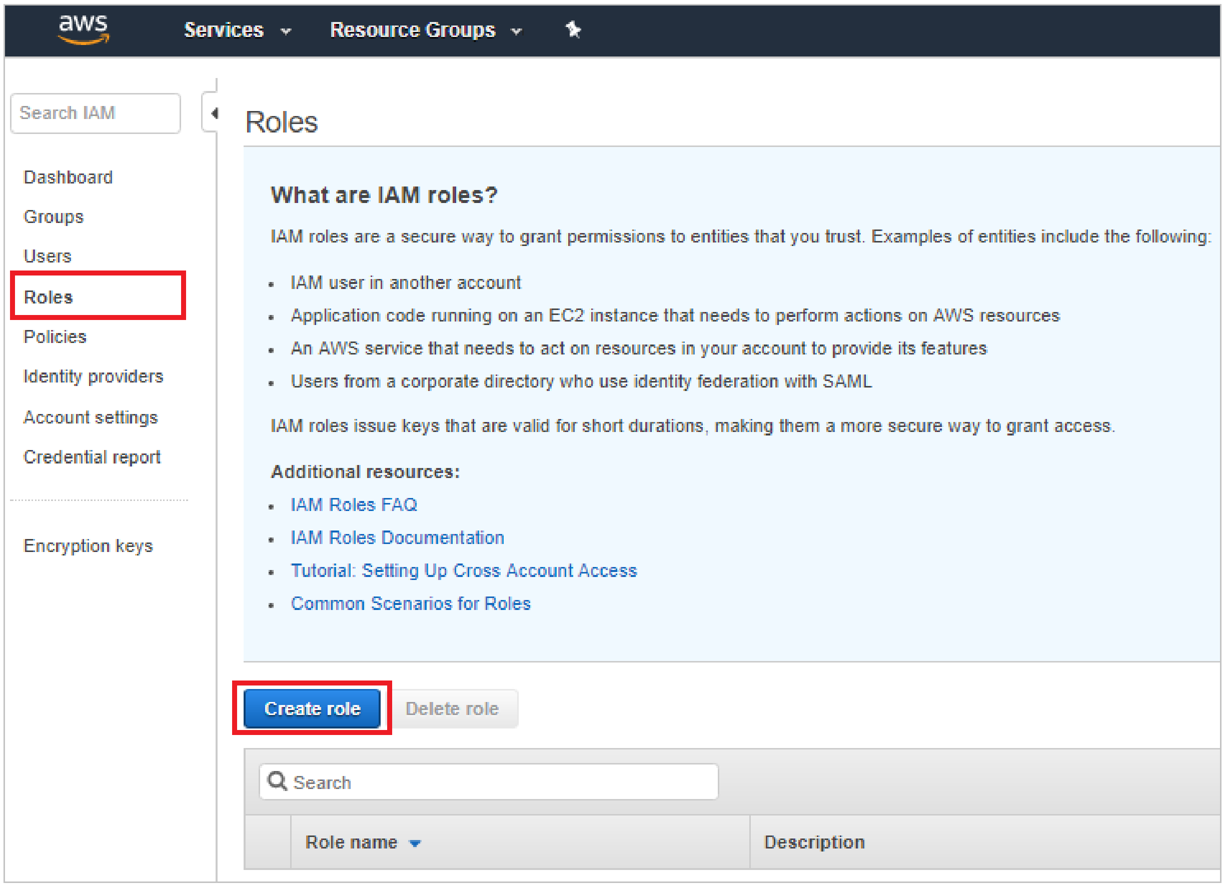
Verify Provider Information

Verify the following provider information. Click **Create** to finish.

Provider Name WAAD

Type SAML

7. On the left pane, select **Roles**, and then select **Create role**.



Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

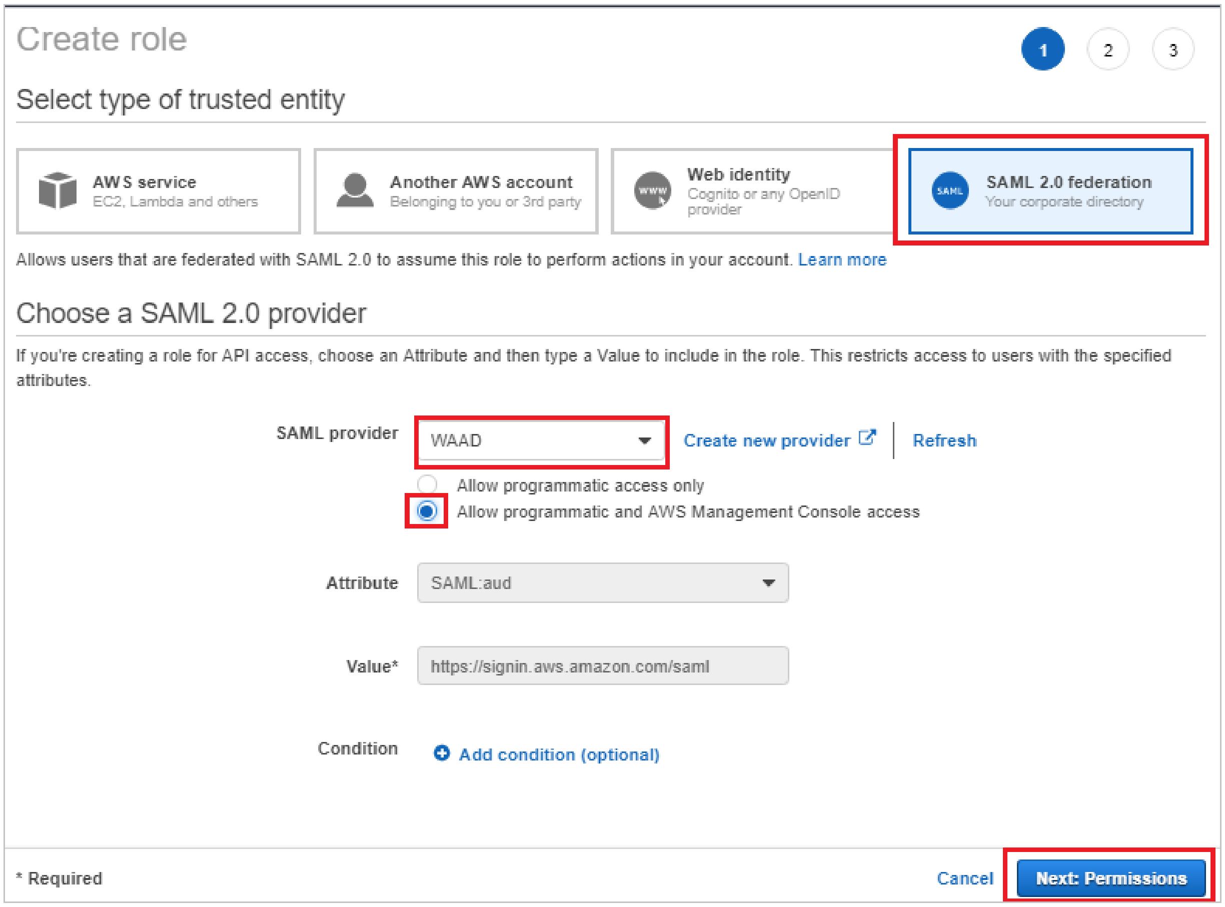
Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

ⓘ Note

The combined length of the role Amazon Resource Name (ARN) and the SAML provider ARN for a role that's being imported must be 240 or fewer characters.

8. On the **Create role** page, do the following:



Select type of trusted entity

Allows users that are federated with SAML 2.0 to assume this role to perform actions in your account.

Choose a SAML 2.0 provider

If you're creating a role for API access, choose an Attribute and then type a Value to include in the role. This restricts access to users with the specified attributes.

SAML provider: WAAD

○ Allow programmatic access only

● Allow programmatic and AWS Management Console access

Attribute: SAML:aud

Value*: https://signin.aws.amazon.com/saml

Condition

* Required

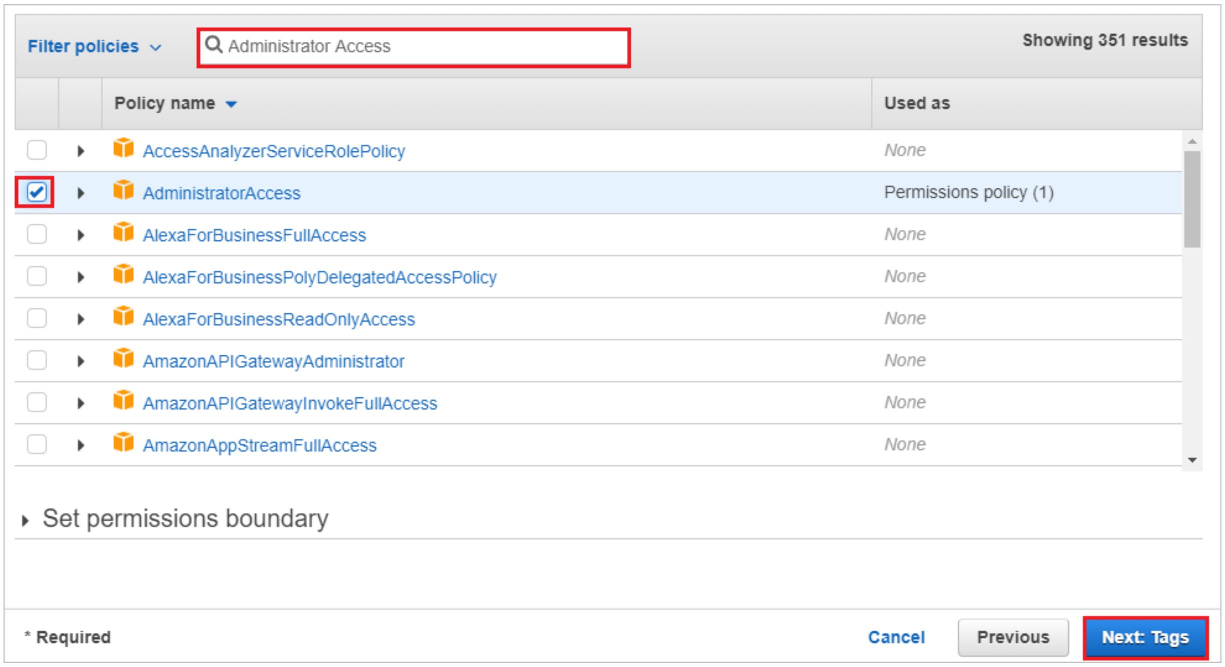
a. Under **Select type of trusted entity**, select **SAML 2.0 federation**.

b. Under **Choose a SAML 2.0 provider**, select the SAML provider that you created previously (for example, *WAAD*)

c. Select **Allow programmatic and AWS Management Console access**.

d. Select **Next: Permissions**.

9. In the search box, enter **Administrator Access**, select the **AdministratorAccess** check box, and then select **Next: Tags**.



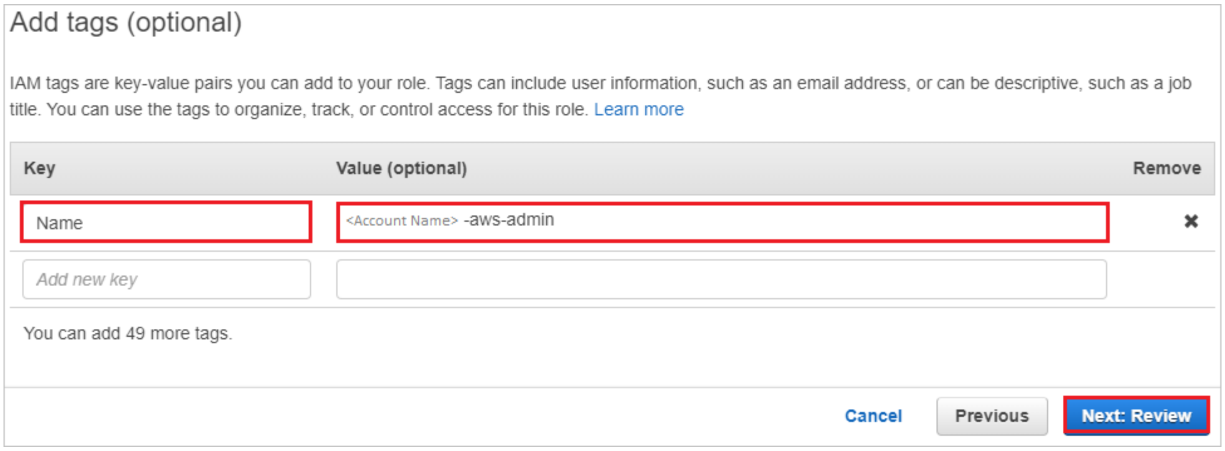
Administrator Access

Showing 351 results

		Policy name	Used as
□	▸	AccessAnalyzerServiceRolePolicy	*None*
☑	▸	AdministratorAccess	Permissions policy (1)
□	▸	AlexaForBusinessFullAccess	*None*
□	▸	AlexaForBusinessPolyDelegatedAccessPolicy	*None*
□	▸	AlexaForBusinessReadOnlyAccess	*None*
□	▸	AmazonAPIGatewayAdministrator	*None*
□	▸	AmazonAPIGatewayInvokeFullAccess	*None*
□	▸	AmazonAppStreamFullAccess	*None*

* Required

10. On the **Add tags (optional)** pane, do the following:



Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. Learn more

Key	Value (optional)	Remove
Name	<Account Name> -aws-admin	✖
Add new key		

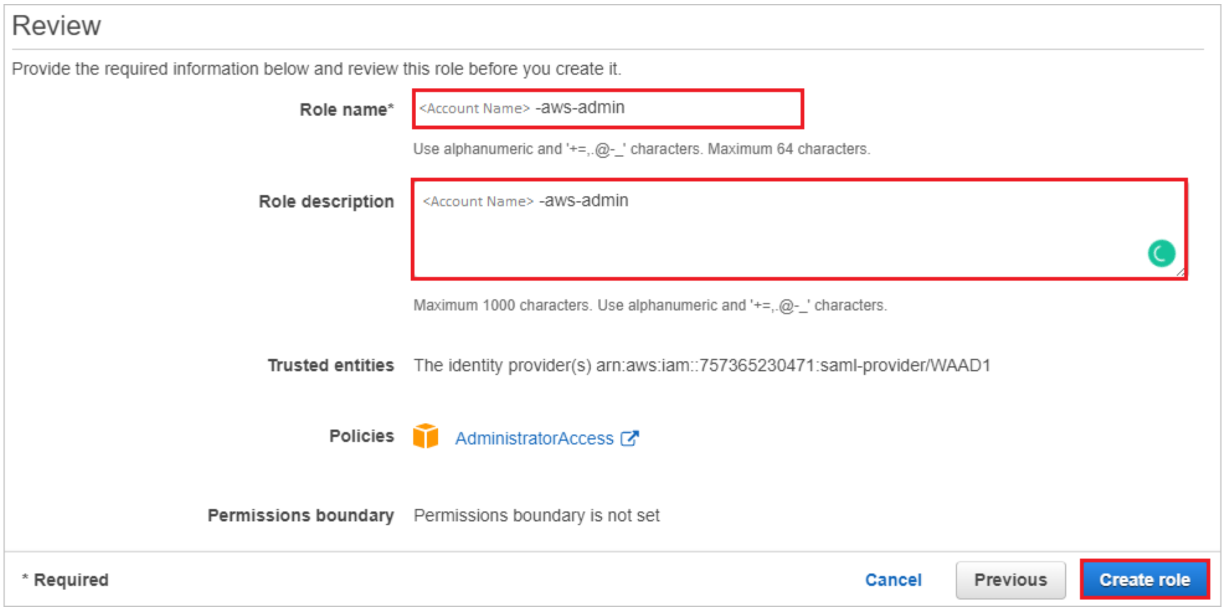
You can add 49 more tags.

a. In the **Key** box, enter the key name (for example, *Azureadtest*).

b. In the **Value (optional)** box, enter the key value in the following format: `<accountname-aws-admin>`. The account name should be in all lowercase letters.

c. Select **Next: Review**.

11. On the **Review** pane, do the following:



Review

Provide the required information below and review this role before you create it.

Role name* <Account Name> -aws-admin

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

Role description <Account Name> -aws-admin

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

Trusted entities The identity provider(s) arn:aws:iam::757365230471:saml-provider/WAAD1

Policies AdministratorAccess

Permissions boundary Permissions boundary is not set

a. In the **Role name** box, enter the value in the following format: `<accountname-aws-admin>`.

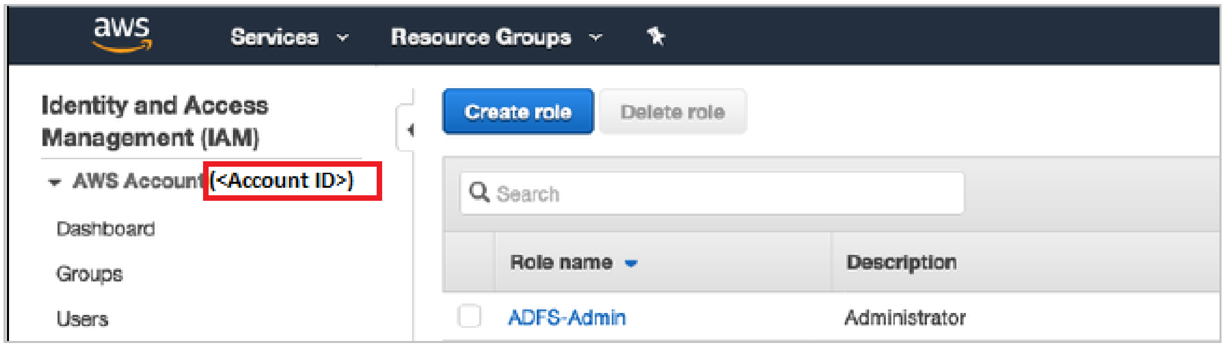
b. In the **Role description** box, enter the value that you used for the role name.

c. Select **Create role**.

d. Create as many roles as you need, and map them to the identity provider.

> ⓘ Note
>
> Similarly, you can create other roles, such as *accountname-finance-admin*, *accountname-read-only-user*, *accountname-devops-user*, or *accountname-tpm-user*, each with a different policy attached to it. You can change these role policies later, according to the requirements for each AWS account. It's a good idea to keep the same policies for each role across the AWS accounts.

12. Be sure to note the account ID for the AWS account either from the Amazon Elastic Compute Cloud (Amazon EC2) properties pane or the IAM dashboard, as shown in the following screenshot:



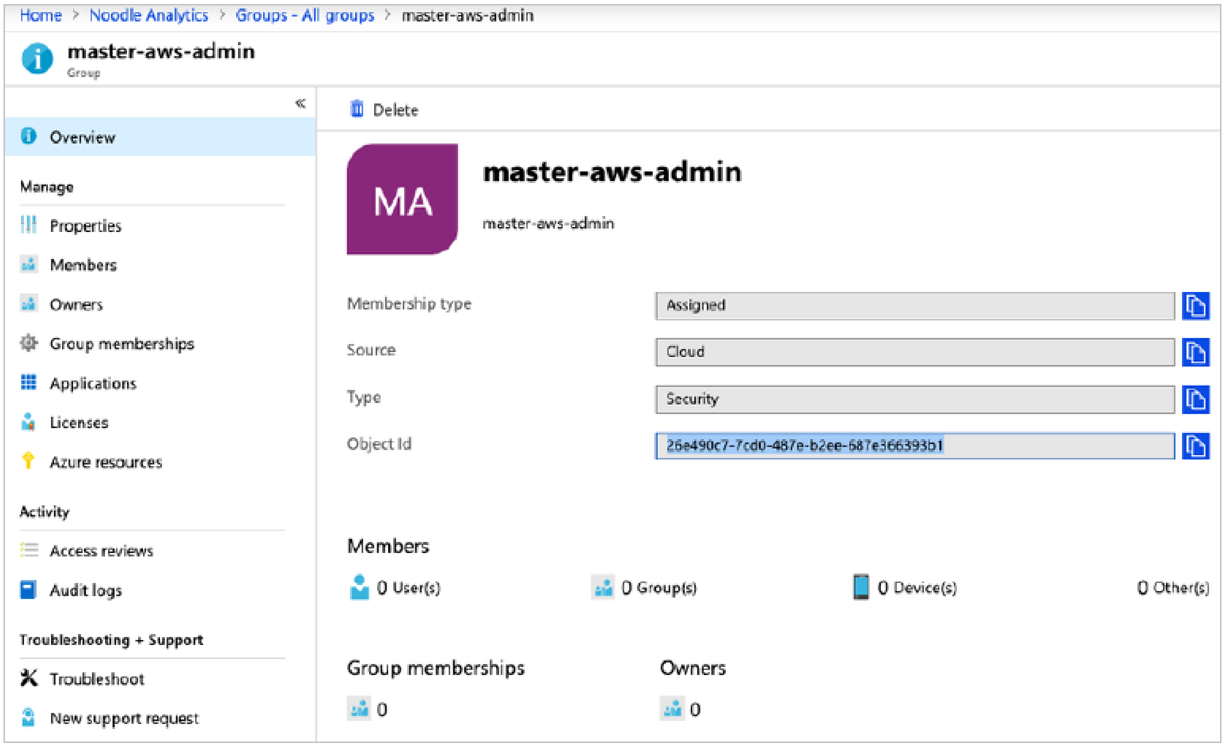
Identity and Access Management (IAM)

AWS Account (<Account ID>)

	Role name	Description
□	ADFS-Admin	Administrator

13. Sign in to the Azure portal, and then go to **Groups**.

14. Create new groups with the same name as that of the IAM roles you created earlier, and then note the value in the **Object Id** box of each of these new groups.



master-aws-admin

master-aws-admin

master-aws-admin

Membership type	Assigned
Source	Cloud
Type	Security
Object Id	26e490c7-7cd0-487e-b2ee-687e366393b1

Members

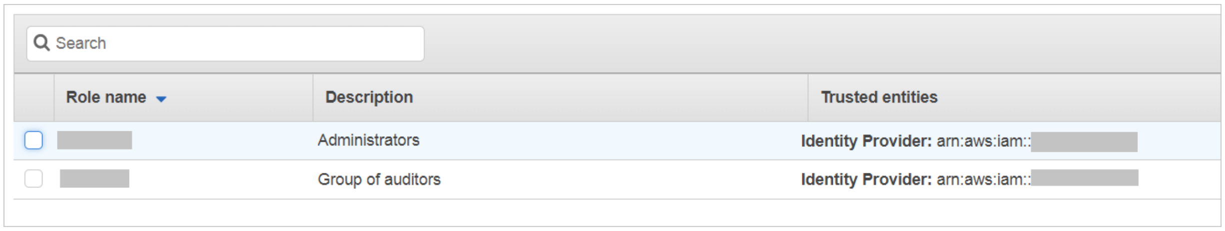
0 User(s) 0 Group(s) 0 Device(s) 0 Other(s)

Group memberships 0

Owners 0

15. Sign out of the current AWS account, and then sign in to another account where you want to configure SSO with Azure AD.

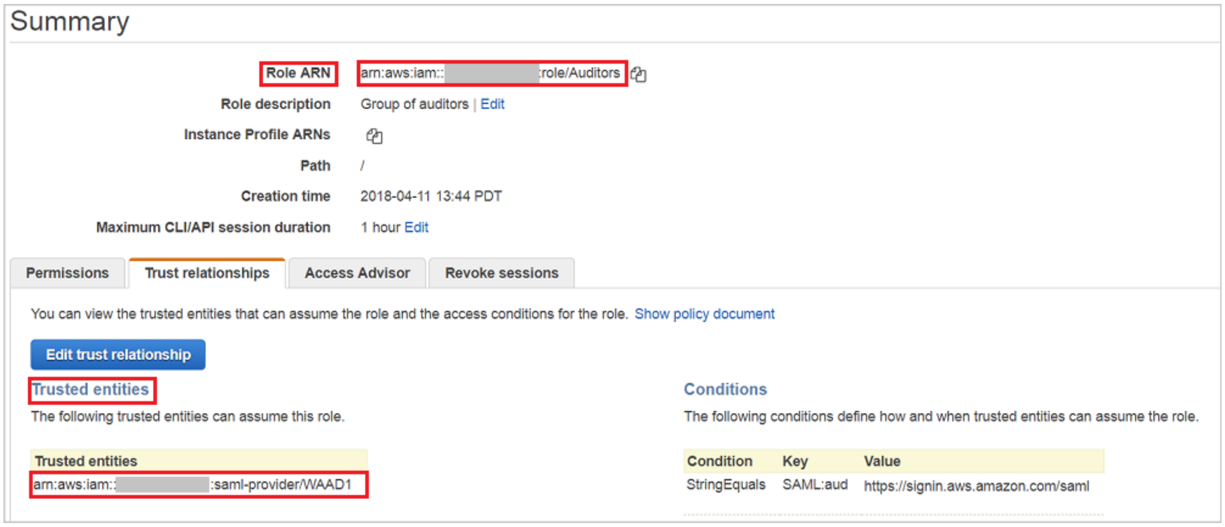
16. After you've created all the roles in the accounts, they're displayed in the **Roles** list for those accounts.



	Role name	Description	Trusted entities
□	[redacted]	Administrators	Identity Provider: arn:aws:iam::[redacted]
□	[redacted]	Group of auditors	Identity Provider: arn:aws:iam::[redacted]

You next need to capture all the role ARNs and trusted entities for all roles across all accounts. You'll need to map them manually with the Azure AD application. To do so:

1. Select each role to copy its role ARN and trusted entity values. You'll need them for all the roles that you'll create in Azure AD.



Summary

Role ARN	arn:aws:iam::[redacted]:role/Auditors
Role description	Group of auditors | Edit
Instance Profile ARNs	
Path	/
Creation time	2018-04-11 13:44 PDT
Maximum CLI/API session duration	1 hour Edit

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

Trusted entities

The following trusted entities can assume this role.

Trusted entities
arn:aws:iam::[redacted]:saml-provider/WAAD1

Conditions

The following conditions define how and when trusted entities can assume the role.

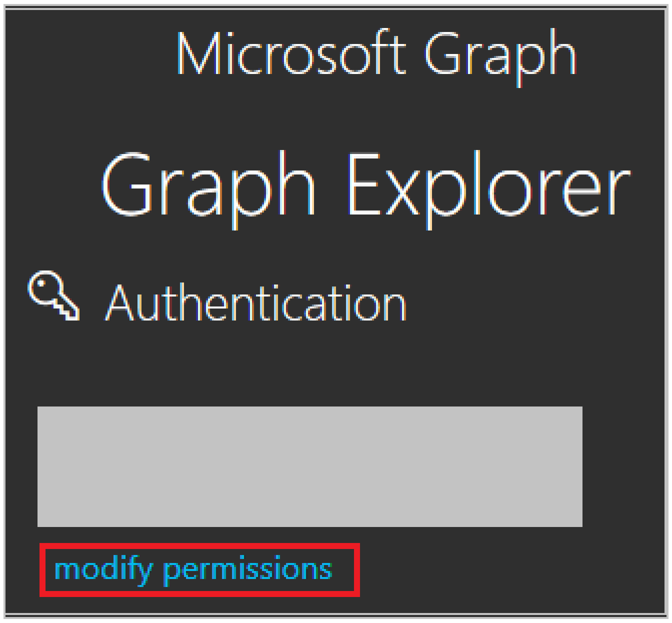
Condition	Key	Value
StringEquals	SAML:aud	https://signin.aws.amazon.com/saml

2. Repeat the preceding step for all the roles in all the accounts, and then store them in a text file in the following format: `<Role ARN>,<Trusted entities>`.

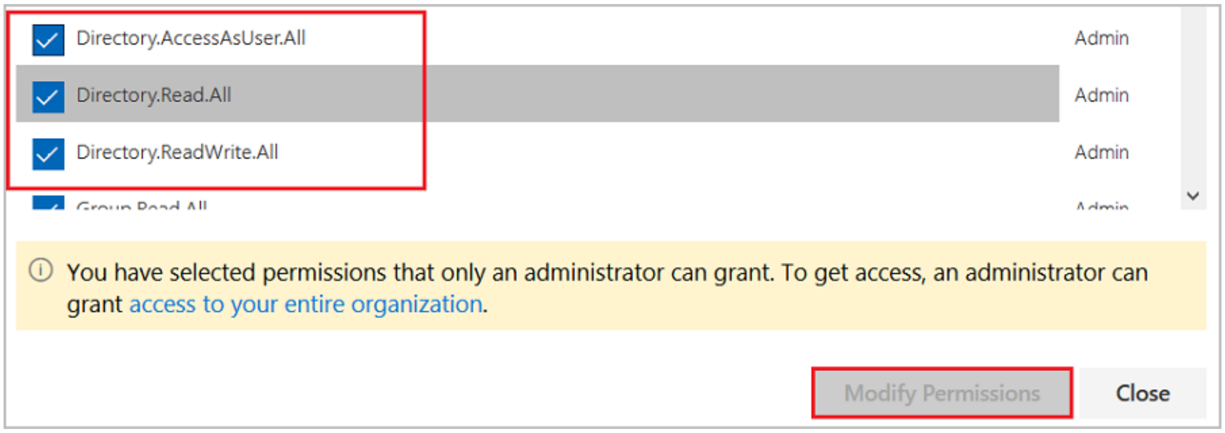
3. Open Microsoft Graph Explorer, and then do the following:

   a. Sign in to the Microsoft Graph Explorer site with the Global Admin or Co-admin credentials for your tenant.

   b. You need sufficient permissions to create the roles. Select **modify permissions**.



c. In the permissions list, if you don't already have the permissions that are shown in the following screenshot, select each one, and then select **Modify Permissions**.



☑ Directory.AccessAsUser.All Admin

☑ Directory.Read.All Admin

☑ Directory.ReadWrite.All Admin

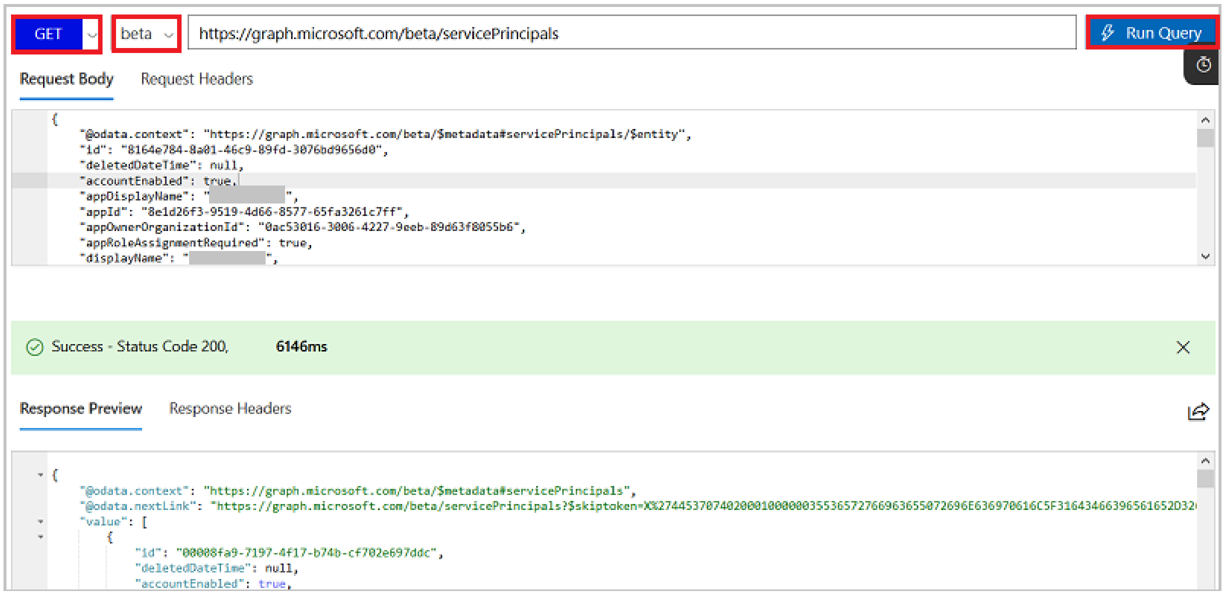
☑ Group.Read.All Admin

ⓘ You have selected permissions that only an administrator can grant. To get access, an administrator can grant access to your entire organization.

d. Sign in to Graph Explorer again, and accept the site usage conditions.

e. At the top of the pane, select **GET** for the method, select **beta** for the version, and then, in the query box, enter either of the following:

- To fetch all the service principals from your tenant, use `https://graph.microsoft.com/beta/servicePrincipals`.
- If you're using multiple directories, use `https://graph.microsoft.com/beta/contoso.com/servicePrincipals`, which contains your primary domain.



https://graph.microsoft.com/beta/servicePrincipals

```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#servicePrincipals/$entity",
    "id": "8164e784-8a01-46c9-89fd-3076bd9656d0",
    "deletedDateTime": null,
    "accountEnabled": true,
    "appDisplayName": "[redacted]",
    "appId": "8e1d26f3-9519-4d66-8577-65fa3261c7ff",
    "appOwnerOrganizationId": "0ac53016-3006-4227-9eeb-89d63f8055b6",
    "appRoleAssignmentRequired": true,
    "displayName": "[redacted]",
```

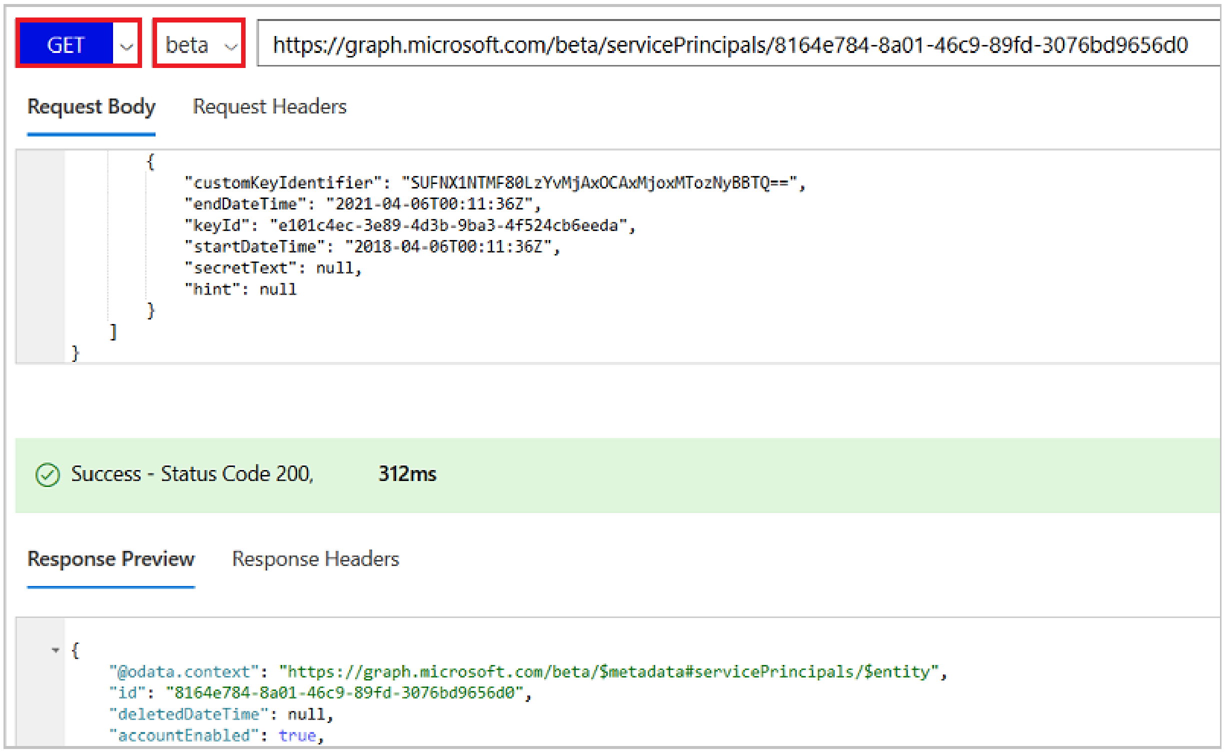
Success - Status Code 200, 6146ms

```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#servicePrincipals",
    "@odata.nextLink": "https://graph.microsoft.com/beta/servicePrincipals?$skiptoken=X%274453707402000100000035536572766963655072696E636970616C5F31643466396561652D32
    "value": [
        {
            "id": "00008fa9-7197-4f17-b74b-cf702e697ddc",
            "deletedDateTime": null,
            "accountEnabled": true,
```

f. From the list of service principals, get the one you need to modify.

You can also search the application for all the listed service principals by selecting Ctrl+F. To get a specific service principal, include in the query the service principal object ID, which you copied earlier from the Azure AD Properties pane, as shown here:

`https://graph.microsoft.com/beta/servicePrincipals/<objectID>`.


GET
beta
https://graph.microsoft.com/beta/servicePrincipals/8164e784-8a01-46c9-89fd-3076bd9656d0
Request Body
Request Headers

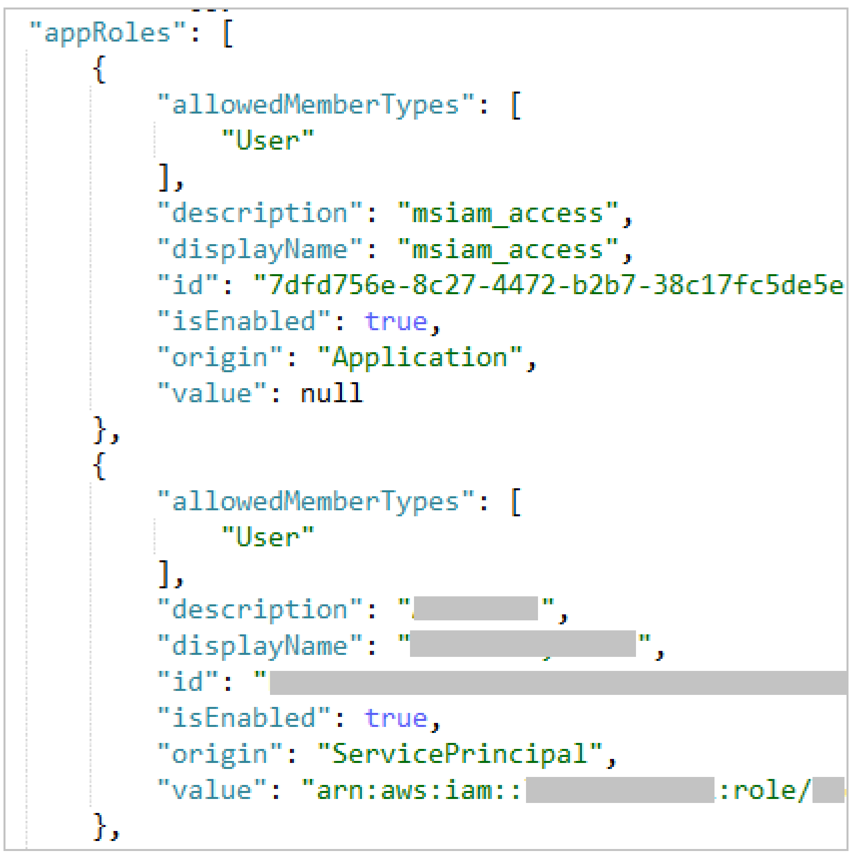
```
                {
                    "customKeyIdentifier": "SUFNX1NTMF80LzYvMjAxOCAxMjoxMTozNyBBTQ==",
                    "endDateTime": "2021-04-06T00:11:36Z",
                    "keyId": "e101c4ec-3e89-4d3b-9ba3-4f524cb6eeda",
                    "startDateTime": "2018-04-06T00:11:36Z",
                    "secretText": null,
                    "hint": null
                }
            ]
        }
```

Success - Status Code 200, 312ms

Response Preview
Response Headers

```
{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#servicePrincipals/$entity",
    "id": "8164e784-8a01-46c9-89fd-3076bd9656d0",
    "deletedDateTime": null,
    "accountEnabled": true,
```

g. Extract the appRoles property from the service principal object.



```
"appRoles": [
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "msiam_access",
        "displayName": "msiam_access",
        "id": "7dfd756e-8c27-4472-b2b7-38c17fc5de5e
        "isEnabled": true,
        "origin": "Application",
        "value": null
    },
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "[redacted]",
        "displayName": "[redacted]",
        "id": "[redacted]
        "isEnabled": true,
        "origin": "ServicePrincipal",
        "value": "arn:aws:iam::[redacted]:role/[redacted]
    },
```

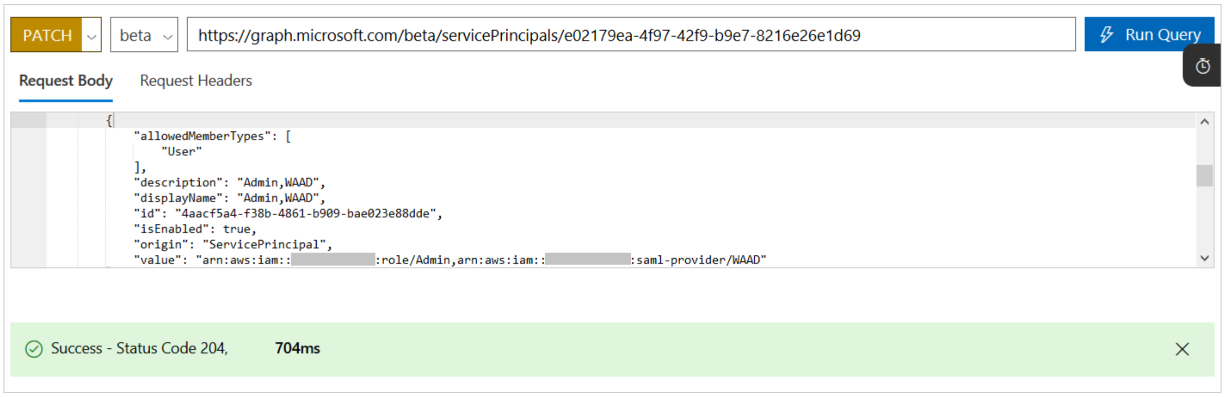
h. You now need to generate new roles for your application.

i. The following JSON code is an example of an appRoles object. Create a similar object to add the roles you want for your application.

```
{
"appRoles": [
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "msiam_access",
        "displayName": "msiam_access",
        "id": "7dfd756e-8c27-4472-b2b7-38c17fc5de5e",
        "isEnabled": true,
        "origin": "Application",
        "value": null
    },
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "Admin,WAAD",
        "displayName": "Admin,WAAD",
        "id": "4aacf5a4-f38b-4861-b909-bae023e88dde",
        "isEnabled": true,
        "origin": "ServicePrincipal",
        "value":
"arn:aws:iam::12345:role/Admin,arn:aws:iam::12345:saml-
provider/WAAD"
    },
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "Auditors,WAAD",
        "displayName": "Auditors,WAAD",
        "id": "bcad6926-67ec-445a-80f8-578032504c09",
        "isEnabled": true,
        "origin": "ServicePrincipal",
        "value":
"arn:aws:iam::12345:role/Auditors,arn:aws:iam::12345:saml-
provider/WAAD"
    }    ]
}
```

> ⓘ Note
>
> You can add new roles only after you've added *msiam_access* for the patch operation. You can also add as many roles as you want, depending on your organization's needs. Azure AD sends the *value* of these roles as the claim value in the SAML response.

j. In Microsoft Graph Explorer, change the method from **GET** to **PATCH**. Patch the service principal object with the roles you want by updating the appRoles property, like the one shown in the preceding example. Select **Run Query** to execute the patch operation. A success message confirms the creation of the role for your AWS application.



https://graph.microsoft.com/beta/servicePrincipals/e02179ea-4f97-42f9-b9e7-8216e26e1d69

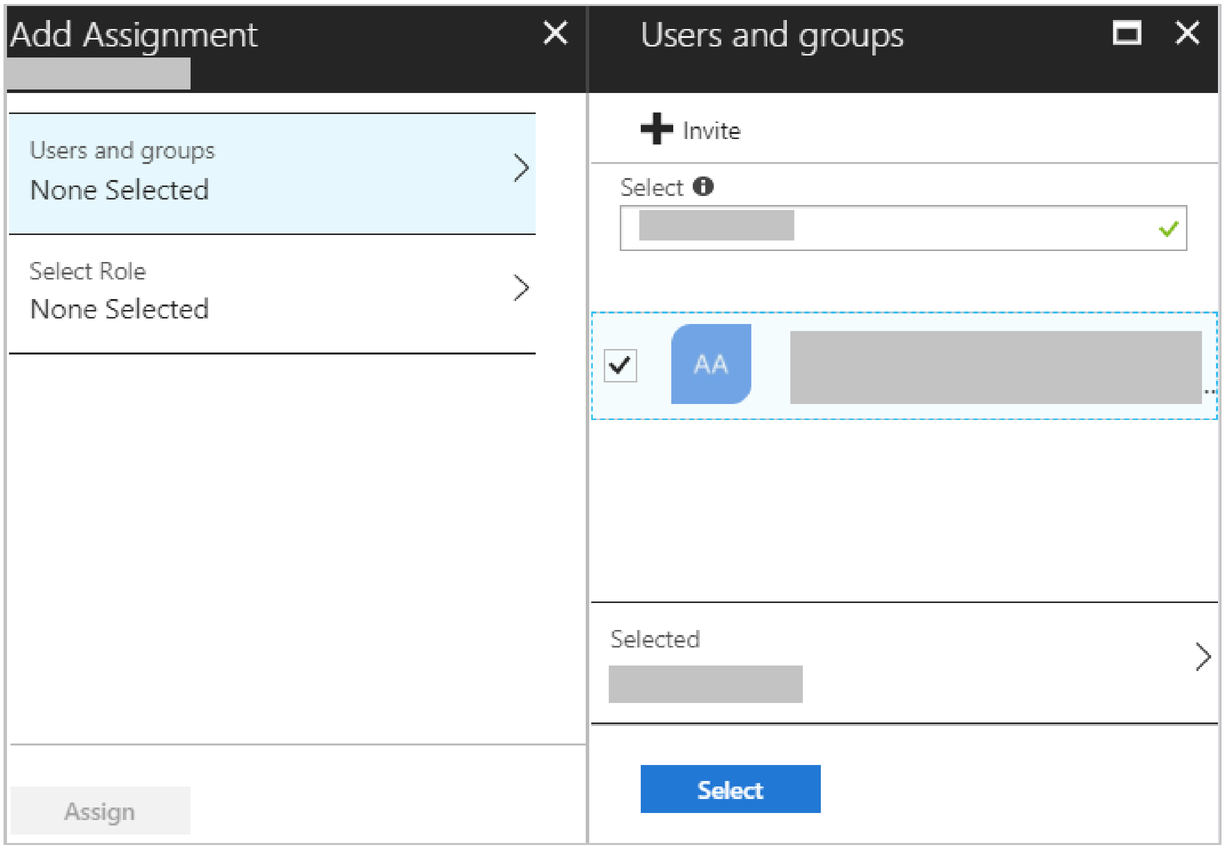
```
{
    "allowedMemberTypes": [
        "User"
    ],
    "description": "Admin,WAAD",
    "displayName": "Admin,WAAD",
    "id": "4aacf5a4-f38b-4861-b909-bae023e88dde",
    "isEnabled": true,
    "origin": "ServicePrincipal",
    "value": "arn:aws:iam::[redacted]:role/Admin,arn:aws:iam::[redacted]:saml-provider/WAAD"
```

Success - Status Code 204, 704ms

4. After the service principal is patched with more roles, you can assign users and groups to their respective roles. You do this in the Azure portal by going to the AWS application and then selecting the **Users and Groups** tab at the top.

5. We recommend that you create a new group for every AWS role so that you can assign that particular role in the group. This one-to-one mapping means that one group is assigned to one role. You can then add the members who belong to that group.

6. After you've created the groups, select the group and assign it to the application.



Add Assignment

[redacted]

Users and groups
None Selected

Select Role
None Selected

Assign

Users and groups

Invite

Select
[redacted]

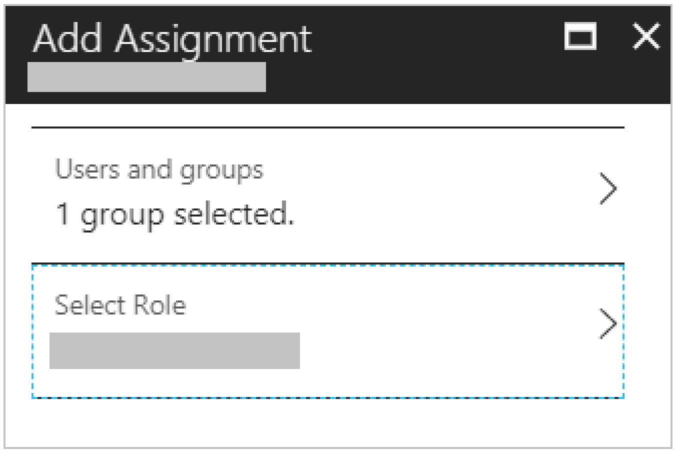
☑ AA [redacted]

Selected
[redacted]

Select

> **Note**
>
> Nested groups are not supported when you assign groups.

7. To assign the role to the group, select the role, and then select **Assign**.



Add Assignment

[redacted]

Users and groups
1 group selected.

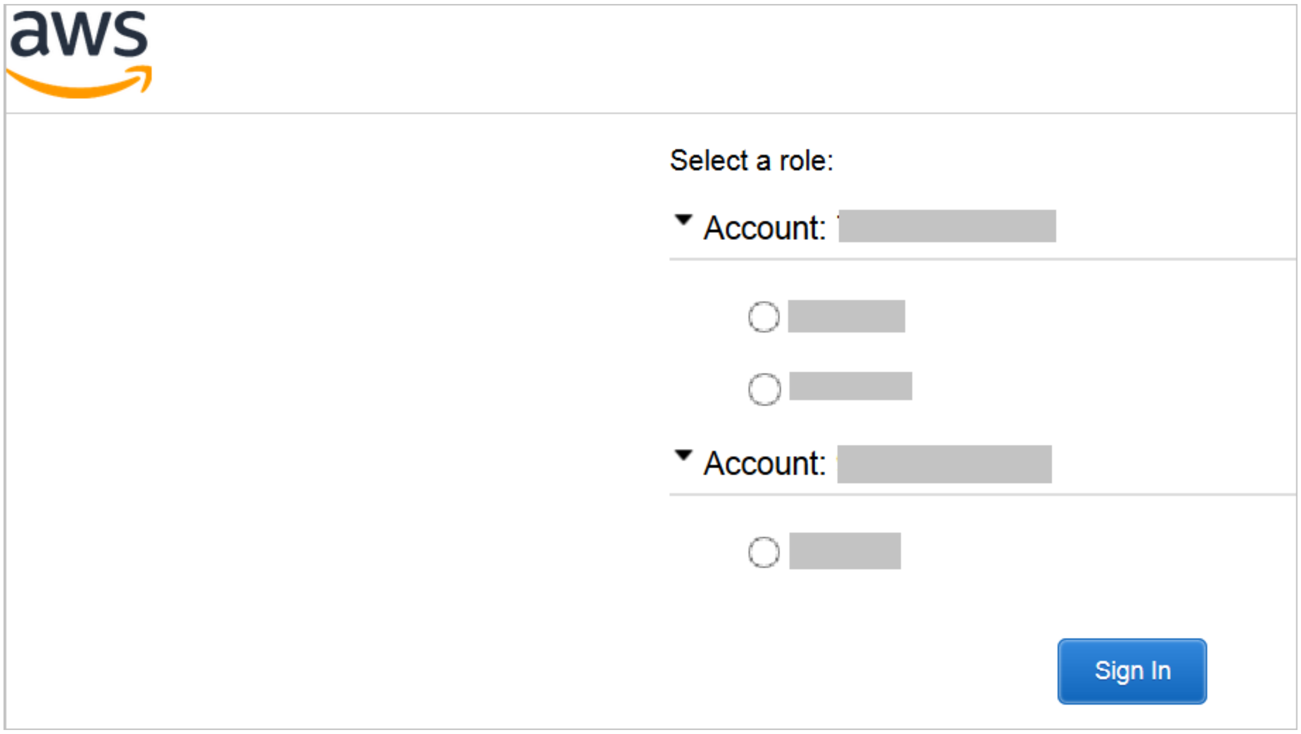
Select Role
[redacted]

> **Note**
>
> After you've assigned the roles, you can view them by refreshing your Azure portal session.

Test SSO

In this section, you test your Azure AD single sign-on configuration by using Microsoft My Apps.

When you select the **AWS** tile in My Apps, the AWS application page opens with an option to select the role.



Select a role:

▾ Account: [redacted]

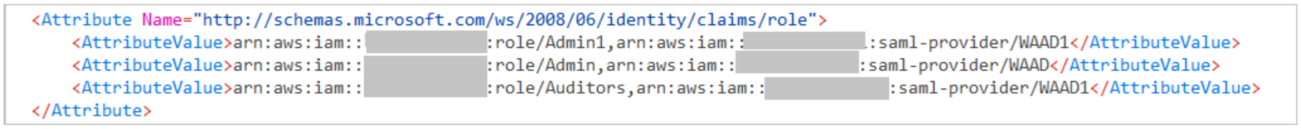
○ [redacted]

○ [redacted]

▾ Account: [redacted]

○ [redacted]

You can also verify the SAML response to see the roles being passed as claims.



```
<Attribute Name="http://schemas.microsoft.com/ws/2008/06/identity/claims/role">
    <AttributeValue>arn:aws:iam::[redacted]:role/Admin1,arn:aws:iam::[redacted]:saml-provider/WAAD1</AttributeValue>
    <AttributeValue>arn:aws:iam::[redacted]:role/Admin,arn:aws:iam::[redacted]:saml-provider/WAAD</AttributeValue>
    <AttributeValue>arn:aws:iam::[redacted]:role/Auditors,arn:aws:iam::[redacted]:saml-provider/WAAD1</AttributeValue>
</Attribute>
```

For more information about My Apps, see Sign in and start apps from the My Apps portal .

Next steps

After you configure AWS you can enforce session control, which protects the exfiltration and infiltration of your organization's sensitive data in real time. Session control extends from conditional access. For more information, see Learn how to enforce session control with Microsoft Defender for Cloud Apps.

Recommended content

Tutorial: Azure AD SSO integration with AWS Single-Account Access

Learn how to configure single sign-on between Azure Active Directory and AWS Single-Account Access.

Tutorial: Azure AD SSO integration with AWS Single Sign-on

Learn how to configure single sign-on between Azure Active Directory and AWS Single Sign-on.

Tutorial: Configure AWS Single Sign-On for automatic user provisioning with Azure Active Directory

Learn how to automatically provision and de-provision user accounts from Azure AD to AWS Single Sign-On.

Tutorial: Azure Active Directory single sign-on (SSO) integration with ServiceNow

Learn how to configure single sign-on between Azure Active Directory and ServiceNow.

Tutorial: Azure Active Directory single sign-on (SSO) integration with JIRA SAML SSO by Microsoft

Learn how to configure single sign-on between Azure Active Directory and JIRA SAML SSO by Microsoft.

Tutorial: Azure Active Directory single sign-on (SSO) integration with Slack

Learn how to configure single sign-on between Azure Active Directory and Slack.

Tutorial: Configure Atlassian Cloud for automatic user provisioning with Azure Active Directory

Learn how to configure Azure Active Directory to automatically provision and de-provision user accounts to Atlassian Cloud.

SAML authentication with Azure Active Directory

Architectural guidance on achieving SAML authentication with Azure Active Directory

Show more