Enabling SAML 2.0 federated users to access the AWS Management Console

PDF (iam-ug.pdf#id_roles_providers_enable-console-saml) | **RSS (aws-iam-release-notes.rss)**

You can use a role to configure your SAML 2.0-compliant identity provider (IdP) and AWS to permit your federated users to access the AWS Management Console. The role grants the user permissions to carry out tasks in the console. If you want to give SAML federated users other ways to access AWS, see one of these topics:

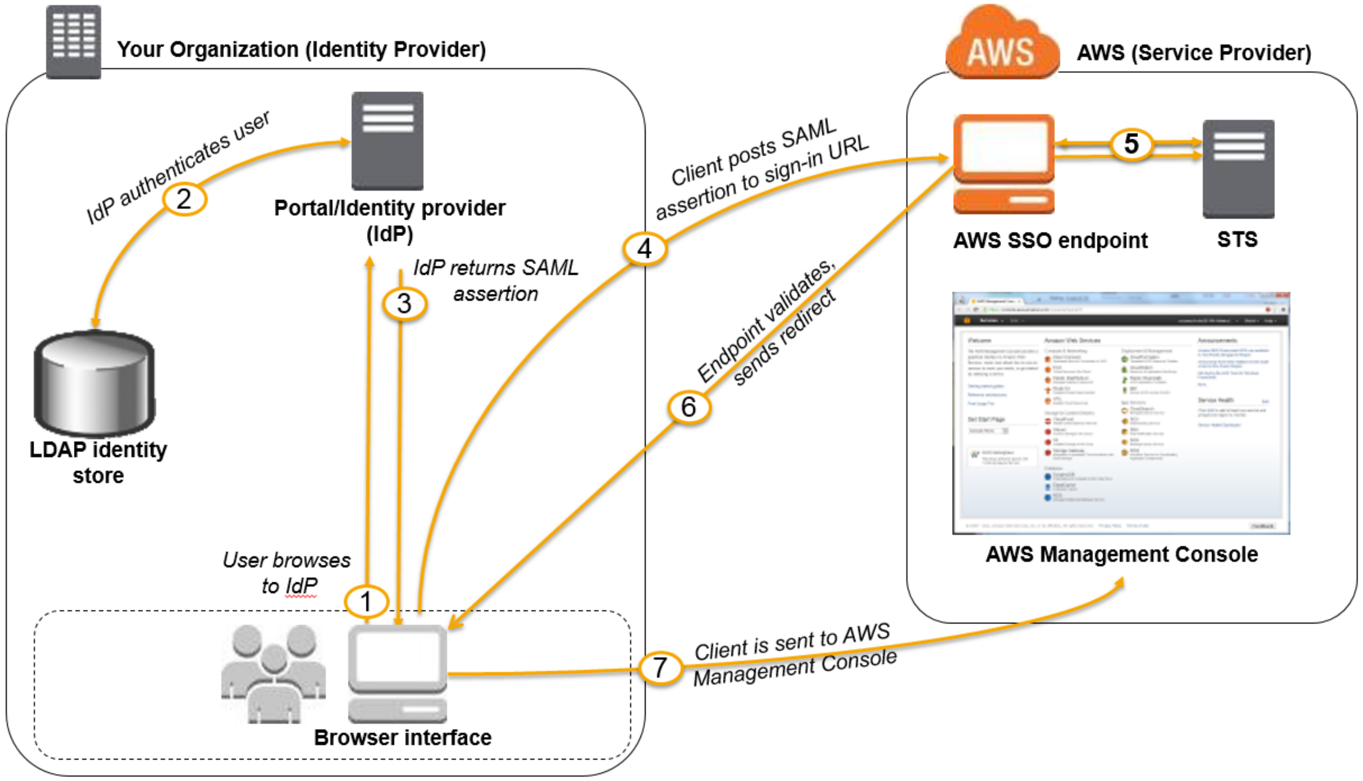
- AWS CLI: Switching to an IAM role (AWS CLI) (./id_roles_use_switch-role-cli.html)
- Tools for Windows PowerShell: Switching to an IAM role (Tools for Windows PowerShell) (./id_roles_use_switch-role-twp.html)
- AWS API: Switching to an IAM role (AWS API) (./id_roles_use_switch-role-api.html)

Overview

The following diagram illustrates the flow for SAML-enabled single sign-on.

Note

This specific use of SAML differs from the more general one illustrated at About SAML 2.0-based federation (./id_roles_providers_saml.html) because this workflow opens the AWS Management Console on behalf of the user. This requires the use of the AWS SSO endpoint instead of directly calling the `AssumeRoleWithSAML` API. The endpoint calls the API for the user and returns a URL that automatically redirects the user's browser to the AWS Management Console.


Your Organization (Identity Provider)
AWS
AWS (Service Provider)
IdP authenticates user
2
Portal/Identity provider (IdP)
IdP returns SAML assertion
3
LDAP identity store
Client posts SAML assertion to sign-in URL
4
5
AWS SSO endpoint
STS
Endpoint validates, sends redirect
6
AWS Management Console
User browses to IdP
1
Client is sent to AWS Management Console
7
Browser interface

The diagram illustrates the following steps:

1. The user browses to your organization's portal and selects the option to go to the AWS Management Console. In your organization, the portal is typically a function of your IdP that handles the exchange of trust between your organization and AWS. For example, in Active Directory Federation Services, the portal URL is: `https://ADFSServiceName/adfs/ls/IdpInitiatedSignOn.aspx`
2. The portal verifies the user's identity in your organization.
3. The portal generates a SAML authentication response that includes assertions that identify the user and include attributes about the user. You can also configure your IdP to include a SAML assertion attribute called `SessionDuration` that specifies how long the console session is valid. You can also configure the IdP to pass attributes as session tags (./id_session-tags.html) . The portal sends this response to the client browser.
4. The client browser is redirected to the AWS single sign-on endpoint and posts the SAML assertion.
5. The endpoint requests temporary security credentials on behalf of the user and creates a console sign-in URL that uses those credentials.
6. AWS sends the sign-in URL back to the client as a redirect.
7. The client browser is redirected to the AWS Management Console. If the SAML authentication response includes attributes that map to multiple IAM roles, the user is first prompted to

select the role for accessing the console.

From the user's perspective, the process happens transparently: The user starts at your organization's internal portal and ends up at the AWS Management Console, without ever having to supply any AWS credentials.

Consult the following sections for an overview of how to configure this behavior along with links to detailed steps.

Configure your network as a SAML provider for AWS

Inside your organization's network, you configure your identity store (such as Windows Active Directory) to work with a SAML-based IdP like Windows Active Directory Federation Services, Shibboleth, etc. Using your IdP, you generate a metadata document that describes your organization as an IdP and includes authentication keys. You also configure your organization's portal to route user requests for the AWS Management Console to the AWS SAML endpoint for authentication using SAML assertions. How you configure your IdP to produce the metadata.xml file depends on your IdP. Refer to your IdP's documentation for instructions, or see Integrating third-party SAML solution providers with AWS (./id_roles_providers_saml_3rd-party.html) for links to the web documentation for many of the SAML providers supported.

Create a SAML provider in IAM

Next, you sign in to the AWS Management Console and go to the IAM console. There you create a new SAML provider, which is an entity in IAM that holds information about your organization's IdP. As part of this process, you upload the metadata document produced by the IdP software in your organization in the previous section. For details, see Creating IAM SAML identity providers (./id_roles_providers_create_saml.html) .

Configure permissions in AWS for your federated users

The next step is to create an IAM role that establishes a trust relationship between IAM and your organization's IdP. This role must identify your IdP as a principal (trusted entity) for purposes of federation. The role also defines what users authenticated by your organization's IdP are allowed to do in AWS. You can use the IAM console to create this role. When you create the trust policy that indicates who can assume the role, you specify the SAML provider that you created earlier in IAM. You also specify one or more SAML attributes that a user must match to be allowed to assume the role. For example, you can specify that only users whose SAML `eduPersonOrgDN (https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_iam-condition-keys.html#ck_edupersonorgdn)` value is `ExampleOrg` are allowed to sign in. The role wizard automatically adds a condition to test the `saml:aud` attribute to make sure that the role is assumed only for sign-in to the AWS Management Console. The trust policy for the role might look like this:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"Federated": "arn:aws:iam::account-id:saml-
provider/ExampleOrgSSOProvider"},
    "Action": "sts:AssumeRoleWithSAML",
    "Condition": {"StringEquals": {
      "saml:edupersonorgdn": "ExampleOrg",
      "saml:aud": "https://signin.aws.amazon.com/saml"
    }}
  }]
}
```

You can include regional endpoints for the `saml:aud` attribute at `https://`*`region-code`*`.signin.aws.amazon.com/static/saml-metadata.xml`. For a list of possible *`region-code`* values, see the **Region** column in AWS Sign-In endpoints (https://docs.aws.amazon.com/general/latest/gr/signin-service.html) .

For the permission policy (./access_policies.html) in the role, you specify permissions as you would for any role, user, or group. For example, if users from your organization are allowed to administer Amazon EC2 instances, you explicitly allow Amazon EC2 actions in the permission policy. You can do this by assigning a managed policy (./access_policies_manage-attach-detach.html) , such as the **Amazon EC2 Full Access** managed policy.

For details about creating a role for a SAML IdP, see Creating a role for SAML 2.0 federation (console) (./id_roles_create_for-idp_saml.html) .

Finish configuration and create SAML assertions

Notify your SAML IdP that AWS is your service provider by installing the `saml-metadata.xml` file found at `https://`*`region-code`*`.signin.aws.amazon.com/static/saml-metadata.xml` or `https://signin.aws.amazon.com/static/saml-metadata.xml`. For a list of possible *`region-code`* values, see the **Region** column in AWS Sign-In endpoints (https://docs.aws.amazon.com/general/latest/gr/signin-service.html) .

How you install that file depends on your IdP. Some providers give you the option to type the URL, whereupon the IdP gets and installs the file for you. Others require you to download the file from the URL and then provide it as a local file. Refer to your IdP documentation for details, or see Integrating third-party SAML solution providers with AWS (./id_roles_providers_saml_3rd-party.html) for links to the web documentation for many of the supported SAML providers.

You also configure the information that you want the IdP to pass as SAML attributes to AWS as part of the authentication response. Most of this information appears in AWS as condition context keys that you can evaluate in your policies. These condition keys ensure that only authorized users in the right contexts are granted permissions to access your AWS resources. You can specify time windows that restrict when the console may be used. You can also specify the maximum time (up to 12 hours) that users can access the console before having to refresh their credentials. For details, see Configuring SAML assertions for the authentication response (./id_roles_providers_create_saml_assertions.html) .