Creating a role for a third-party Identity Provider (federation)

PDF (iam-ug.pdf#id_roles_create_for-idp) | RSS (aws-iam-release-notes.rss)

You can use identity providers instead of creating IAM users in your AWS account. With an identity provider (IdP), you can manage your user identities outside of AWS and give these external user identities permissions to access AWS resources in your account. For more information about federation and identity providers, see Identity providers and federation (./id_roles_providers.html) .

Creating a role for federated users (console)

The procedures for creating a role for federated users depend on your choice of third-party providers:

- For Web Identity or OpenID Connect (OIDC), see Creating a role for web identity or OpenID Connect Federation (console) (./id_roles_create_for-idp_oidc.html) .
- For SAML 2.0, see Creating a role for SAML 2.0 federation (console) (./id_roles_create_for-idp_saml.html) .

Creating a role for federated access (AWS CLI)

The steps to create a role for the supported identity providers (OIDC or SAML) from the AWS CLI are identical. The difference is in the contents of the trust policy that you create in the prerequisite steps. Begin by following the steps in the **Prerequisites** section for the type of provider you are using:

- For an OIDC provider, see Prerequisites for creating a role for web identity or OIDC (./id_roles_create_for-idp_oidc.html#idp_oidc_Prerequisites) .
- For a SAML provider, see Prerequisites for creating a role for SAML (./id_roles_create_for-idp_saml.html#idp_saml_Prerequisites) .

Creating a role from the AWS CLI involves multiple steps. When you use the console to create a role, many of the steps are done for you, but with the AWS CLI you must explicitly perform each step yourself. You must create the role and then assign a permissions policy to the role. Optionally, you can also set the permissions boundary (./access_policies_boundaries.html) for your role.

To create a role for identity federation (AWS CLI)

1. Create a role: aws iam create-role (https://docs.aws.amazon.com/cli/latest/reference/iam/create-role.html)
2. Attach a permissions policy to the role: aws iam attach-role-policy (https://docs.aws.amazon.com/cli/latest/reference/iam/attach-role-policy.html)

   or

   Create an inline permissions policy for the role: aws iam put-role-policy (https://docs.aws.amazon.com/cli/latest/reference/iam/put-role-policy.html)
3. (Optional) Add custom attributes to the role by attaching tags: aws iam tag-role (https://docs.aws.amazon.com/cli/latest/reference/iam/tag-role.html)

   For more information, see Managing tags on IAM roles (AWS CLI or AWS API) (./id_tags_roles.html#id_tags_roles_procs-cli-api) .
4. (Optional) Set the permissions boundary (./access_policies_boundaries.html) for the role: aws iam put-role-permissions-boundary (https://docs.aws.amazon.com/cli/latest/reference/iam/put-role-permissions-boundary.html)

   A permissions boundary controls the maximum permissions that a role can have. Permissions boundaries are an advanced AWS feature.

The following example shows the first two, and most common, steps for creating an identity provider role in a simple environment. This example allows any user in the `123456789012` account to assume the role and view the `example_bucket` Amazon S3 bucket. This example also assumes that you are running the AWS CLI on a computer running Windows, and have already configured the AWS CLI with your credentials. For more information, see Configuring the AWS Command Line Interface (https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html) .

In this example, include the following trust policy in the first command when you create the role. This trust policy allows users in the `123456789012` account to assume the role using the `AssumeRole` operation, but only if the user provides MFA authentication using the `SerialNumber` and `TokenCode` parameters. For more information about MFA, see Using multi-factor authentication (MFA) in AWS (./id_credentials_mfa.html) .

The following example trust policy is designed for a mobile app if the user signs in using Amazon Cognito. In this example, *us-east:12345678-ffff-ffff-ffff-123456* represents the identity pool ID assigned by Amazon Cognito.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "RoleForCognito",
        "Effect": "Allow",
        "Principal": {"Federated": "cognito-
identity.amazonaws.com"},
        "Action": "sts:AssumeRoleWithWebIdentity",
        "Condition": {"StringEquals": {"cognito-
identity.amazonaws.com:aud": "us-east:12345678-ffff-ffff-ffff-
123456"}}
    }
}
```

The following permissions policy allows anyone who assumes the role to perform only the `ListBucket` action on the `example_bucket` Amazon S3 bucket.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}
```

To create this `Test-Cognito-Role` role, you must first save the previous trust policy with the name `trustpolicyforcognitofederation.json` and the previous permissions policy with the name `permspolicyforcognitofederation.json` to the `policies` folder in your local `C:` drive. You can then use the following commands to create the role and attach the inline policy.

```
# Create the role and attach the trust policy that enables users in
an account to assume the role.
$ aws iam create-role --role-name Test-Cognito-Role --assume-role-
```

```
policy-document
file://C:\policies\trustpolicyforcognitofederation.json

# Attach the permissions policy to the role to specify what it is
allowed to do.
aws iam put-role-policy --role-name Test-Cognito-Role --policy-name
Perms-Policy-For-CognitoFederation --policy-document
file://C:\policies\permspolicyforcognitofederation.json
```

Creating a role for federated access (AWS API)

The steps to create a role for the supported identity providers (OIDC or SAML) from the AWS CLI are identical. The difference is in the contents of the trust policy that you create in the prerequisite steps. Begin by following the steps in the **Prerequisites** section for the type of provider you are using:

- For an OIDC provider, see Prerequisites for creating a role for web identity or OIDC (./id_roles_create_for-idp_oidc.html#idp_oidc_Prerequisites) .
- For a SAML provider, see Prerequisites for creating a role for SAML (./id_roles_create_for-idp_saml.html#idp_saml_Prerequisites) .

To create a role for identity federation (AWS API)

1. Create a role: CreateRole (https://docs.aws.amazon.com/IAM/latest/APIReference/API_CreateRole.html)
2. Attach a permissions policy to the role:AttachRolePolicy (https://docs.aws.amazon.com/IAM/latest/APIReference/API_AttachRolePolicy.html)

   or

   Create an inline permissions policy for the role: PutRolePolicy (https://docs.aws.amazon.com/IAM/latest/APIReference/API_PutRolePolicy.html)
3. (Optional) Add custom attributes to the user by attaching tags: TagRole (https://docs.aws.amazon.com/IAM/latest/APIReference/API_TagRole.html)

   For more information, see Managing tags on IAM users (AWS CLI or AWS API) (./id_tags_users.html#id_tags_users_procs-cli-api) .
4. (Optional) Set the permissions boundary (./access_policies_boundaries.html) for the role: PutRolePermissionsBoundary (https://docs.aws.amazon.com/IAM/latest/APIReference/API_PutRolePermissionsBoundary.html)

A permissions boundary controls the maximum permissions that a role can have. Permissions boundaries are an advanced AWS feature.
