Deloitte.

AWS Azure AD SSO Plugin

Standards and Guidelines

September 2019

Document Control Information

Document Author	Deloitte ITS Cloud Services (DCS Deloitte Cloud Services)
Date Released	2019-September

Document Edit History

Version	Date	Additions/Modifications	Prepared/Reviewed By
1	09/2019	Document creation	Scott Judson

Table of Contents

Azure AD SSO Plugin

This plugin will allow access to AWS accounts through your Deloitte privileged accounts. It is required that access goes through privileged accounts managed by Global PCS or member firm PAM solution. Standard active directory accounts cannot be used.

Adding Additional Users

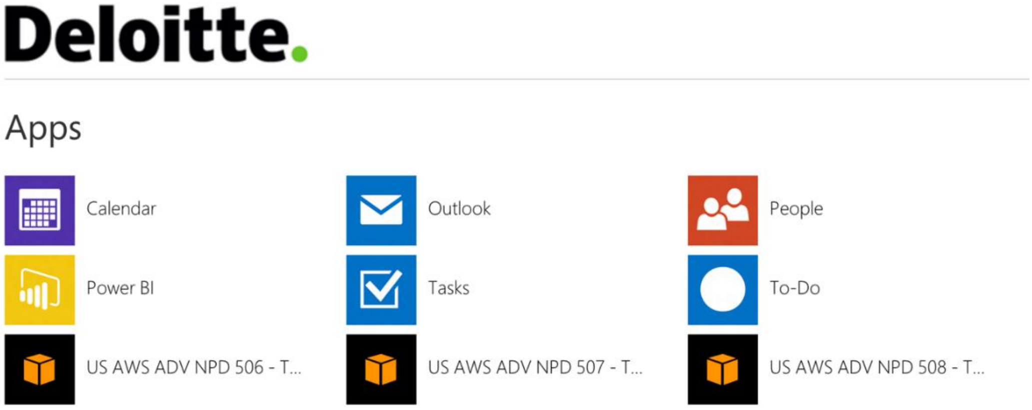
During account provisioning, a set of Active Directory groups are created for access into the account. It typically includes groups for different job functions such as Admin, DBAdmin, PowerUser, Reader, SysAdmin. It is customizable per Manage Service Provider (MSP). The naming convention for these groups are "SG-CountryCode-AccountId-JobFunction". Ie SG-US-111111111111-Admin.

Initially, the SG for the accounts will be empty. The technical administrator will be the owner of all SGs and can add additional privileged accounts as needed. Access for non US member firms can granted through Microsoft Identity Manager (MIM). For US member firm, the technical administrator and other team members can request access via the MAC tool.

Note: A possible two (2) hour delay can occur before access to the newly added AWS account is available.

Accessing Your Account Through Console

- Navigate to https://myapps.microsoft.com. Refer to the notes below for browser specific information with privileged accounts.

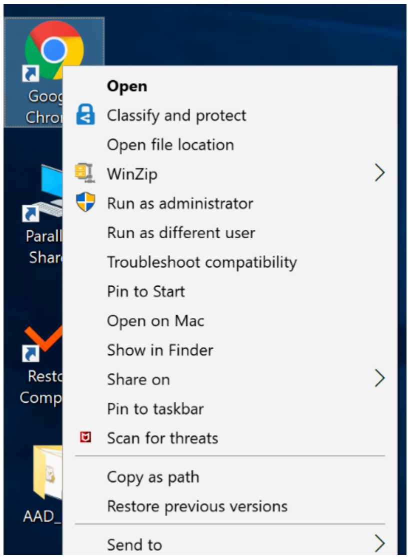

Deloitte.
Apps
Calendar
Outlook
People
Power BI
Tasks
To-Do
US AWS ADV NPD 506 - T...
US AWS ADV NPD 507 - T...
US AWS ADV NPD 508 - T...

- The available AWS accounts will be listed each as an individual tile with the naming convention of "Country Code AWS Function Environment Code – Project Name"

- Select the account. Enter MFA information if prompted. Access to the account will be granted.

Note (Windows): The Firefox browser can be used as is and does not require additional steps. To use Chrome or Internet explorer to access the account, requires launching as a different user as the browsers automatically login using the standard account. Follow the steps below:

- Hold down shift key and right click the browser icon
- Select run as a different user
- Enter the privileged account credentials.



Note (Mac): Use Firefox to access your account.

Accessing Your Account Through Command Line Interface (CLI)

General posture is to leverage Azure AD for CLI access into your accounts which provides you with one-hour temporary access keys. There may be exceptions to this rule that warrant long term access keys. Use cases for the exceptions are available. If you are in the US member firm and require an exception, please submit a Service Now (SNOW) form.

Windows Users

1. Download latest version of node and save it in the root of C: directory
2. Run the installation under the privileged account.

   Hold down shift key and right click the downloaded installation file

   Select Run as different user

   When prompted, enter privileged account credentials

3. Run **cmd** prompt under privileged account. Search for **cmd**, right click, select open file location.

   Hold down shift key and right click the **cmd** prompt shortcut

   Select run as different user

   When prompted, enter privileged account credentials

4. Change to privileged account directory with the cd command:

   ```
   cd C:\Users\usa-jdoe
   ```

5. Install aws-azure-login with npm:

   ```
   npm install -g aws-azure-login
   ```

6. Configure account:

   ```
   aws-azure-login --configure
   ```

   To save configuration with a specific profile name use:

   ```
   aws-azure-login --configure --profile foo
   ```

Azure App ID URI: https://signin.aws.amazon.com/saml#AccountID

Note: Substitute the AccountID above with the account id of the account you are accessing

Default Username: usa-jdoe@deloitte.com

Default Role ARN: Leave blank

Default Session Duration Hours: 1

7. Login:

```
aws-azure-login
```

Enter username and password and MFA information if prompted.

To log in with a named profile:

```
aws-azure-login --profile foo
```

Or set the AWS_PROFILE environmental variable to the name of the profile just like the AWS CLI.

Upon logon, the AWS CLI or SDKs are available for use.

Mac Users

Before starting, download and install the latest version of nod e

Option A: Install for All Users

Install aws-azure-login globally with npm:

```
sudo npm install -g aws-azure-login --unsafe-perm
```

Puppeteer doesn't install globally with execution permissions for all users, so modification is required:

```
sudo chmod -R go+rx $(npm root -g)
```

Continue the configuration by following step 3 below.

Option B: Install Only for Current User

1. Configure npm to install global packages in home directory:

```
mkdir ~/.npm-global

npm config set prefix '~/.npm-global'

export PATH=~/.npm-global/bin:$PATH

source ~/.profile

echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.profile

source ~/.profile
```

2. Install aws-azure-login:

```
npm install -g aws-azure-login
```

3. Configure account:

```
aws-azure-login --configure
```

To save configuration with a specific profile name use:

```
aws-azure-login --configure --profile foo
```

4. Enter the following:

   Azure Tenant ID: 36da45f1-dd2c-4d1f-af13-5abe46b99921

   Azure App ID URI: https://signin.aws.amazon.com/saml#AccountID

   Note: Substitute the AccountID above with the account id of the account you need to access

   Default Username: usa-jdoe@deloitte.com

   Default Role ARN: Leave blank

Default Session Duration Hours: 1

5. Login:

```
aws-azure-login --mode gui
```

Enter username and password in the browser window that opens. If MFA is required, enter the verification code or mobile device approval.

```
aws-azure-login --profile foo --mode gui
```

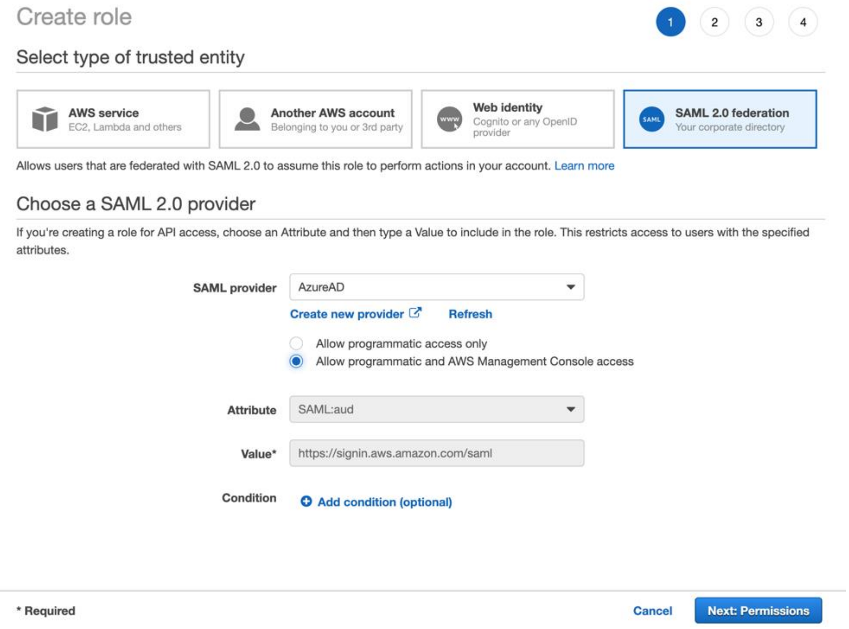
Or set the AWS_PROFILE environmental variable to the name of the profile just like the AWS CLI.

Upon logon, the AWS CLI or SDKs are available for use.

Creating Custom SSO Roles

To create a custom SSO role, within your account create an IAM role named AWS_AccountId_Function. For example, you can create an IAM role with the name AWS_111111111111_S3 and assign only S3 permissions to the role. When creating the role, under “Select type of trusted entity” choose SAML 2.0 federation and choose AzureAD for SAML provider. See screenshot below.

Once the IAM role has been created, this will start an automation job that will automatically create the security group and link the group to your AWS account. The technical administrator will receive an email once the security group is ready to go. This could take up to two (2) hours.



Create role

Select type of trusted entity

- AWS service — EC2, Lambda and others
- Another AWS account — Belonging to you or 3rd party
- Web identity — Cognito or any OpenID provider
- SAML 2.0 federation — Your corporate directory

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

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: AzureAD

○ Allow programmatic access only

● Allow programmatic and AWS Management Console access

Attribute: SAML:aud

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

Condition: Add condition (optional)

* Required

For additional plugin information, reference the AWS plugins.

FAQs

1. How do I get access to my AWS account?

   US member firm - request access to the appropriate account security group within the MAC tool.

   Other member firms - the account admin can grant access through MIM. Please refer to the section Adding Additional Users within this document.

2. I'm unable to see my account in the MyApps portal. What do I need to do?

   Check the following:

   - Make sure you are logged into your privileged account by clicking on your name in the top right-hand corner. You should see your privileged account username. For example, usa-jdoe.

   - Make sure you are a member of the account SGs. For US member firm, you can verify by logging into the MAC tool, select Manage My Access, and then click Remove Access. Search for the account using the account id to make sure you are a member of the account security group

3. I'm getting an error when logging into the MyApps portal. What is the issue?

   US member firm - if you did not have a usa account, once your MAC request is approved it automatically creates a usa account for you. Before you can log into the MyApps portal, this new account needs to sync with Azure AD which could take up to two (2) hours.

   Other member firms - once you have a privileged account created, this will also need to sync with Azure AD in order to log into the MyApps portal

4. I've been added to the appropriate security group. Why can't I see my account in the MyApps portal?

   It could be up to two (2) hours before you are able to see your account in the MyApps portal because the group membership needs to sync with Azure AD.

5. When accessing my account through the CLI, I'm receiving an error message saying that my application was not found in the directory. What is the issue?

   When configuring CLI access, make sure there are no extra spaces in the beginning of the line when you enter the Tenant ID or App ID URI.