Clarity Innovations

Technical Documentation

Configuring Google SSO with Amazon AppStream 2.0 and Amazon AppStream 2.0 Chrome Packaging and Deployment

Version 1 - December 2017

Configuring Google SSO with Amazon AppStream 2.0

Requirements

1. You have an AWS account
2. You have created at least one AppStream stack
3. You have Administrator access to G Suite

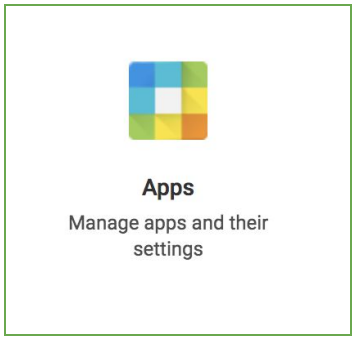
Configuration Steps

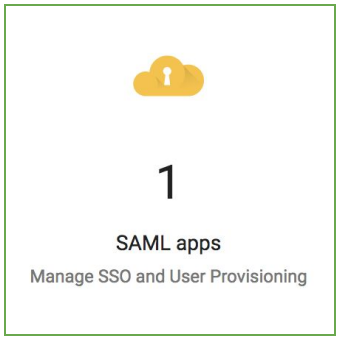
1. Gather your Google Metadata documents
2. Create your Identity Provider
3. Create a Role that uses your Identity Provider
4. Create custom attributes for SAML information
5. Finish creating the SAML application

Get your Google Metadata Documents

1. In your Google Admin console (admin.google.com):

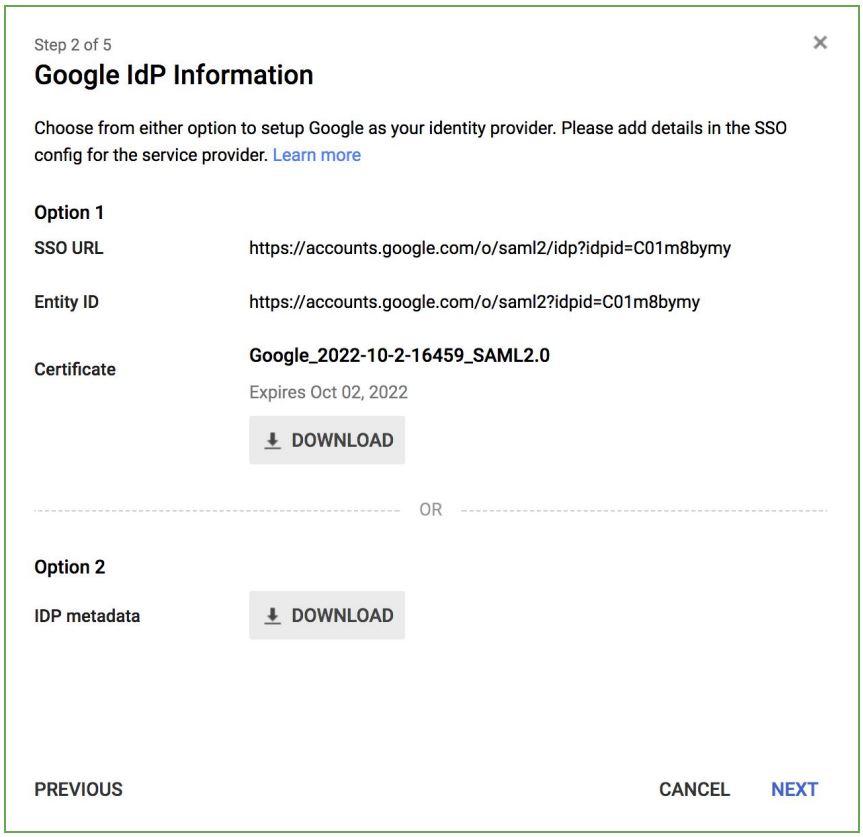
   Navigate to **Apps** -> **SAML apps**





2. Click the **yellow + button** in the bottom-right corner, then choose “Amazon Web Services.”

3. Under **Option 2**, click the Download button to download your IDP metadata. It will download as a file named “GoogleIDPMetadata-*your-domain.edu*.xml.”



4. Keep this tab open for steps later in the process.

Create your Identity Provider

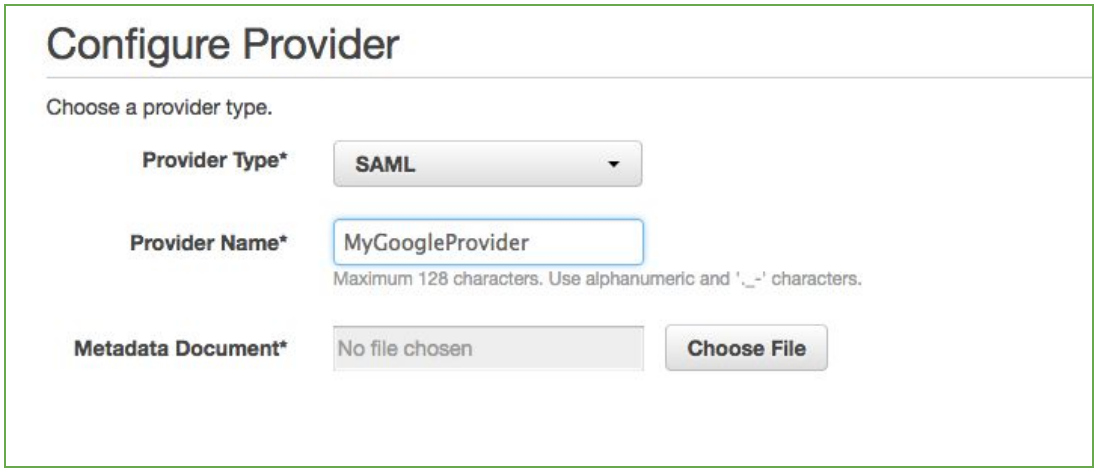
1. In AWS IAM (https://console.aws.amazon.com/iam/), navigate to “Identity providers” using the menu on the left

2. Click **Create Provider**.

3. For **Provider Type**, choose SAML.

4. Enter a **Provider Name** identifying it as a Google provider. You will use this name later in the process.

5. For the **metadata document**, upload the IDP metadata you saved in the last step.

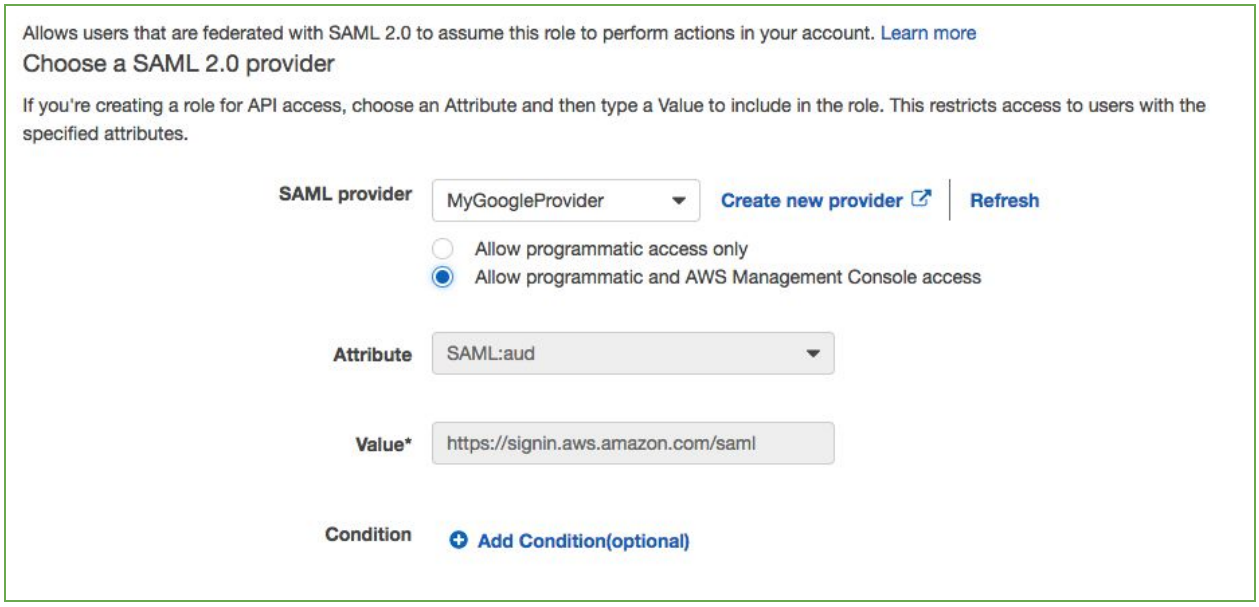


6. On the next screen, verify the provider name and type, then click “**Create**.”

7. Click the Provider Name to bring up the summary. Note the **Provider ARN**, as it will be used in a later step.

Create a Role that uses your Identity Provider

1. In AWS IAM, navigate to **Roles** using the menu on the left.
2. Click **Create role**.
3. You'll see a choice of four types. Choose **Saml 2.0 federation**.
4. In the **SAML provider** drop-down menu, select the Identity Provider you created in the last step.
5. Select **Allow programmatic and AWS Management Console access**. When you do, the other fields will populate automatically.



6. Click **Next: Permissions**.
7. You'll need to create a new permission policy. Click **Create policy**. This will open a new tab.

8. In the new tab, choose **Create Your Own Policy**.

9. Fill in a name (e.g. AppStreamSSOPolicy) and description for the policy.

10. For the Policy document, use this template:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "appstream:Stream",
            "Resource": "<Appstream Stack Resource>",
            "Condition": {
                "StringEquals": {
                    "appstream:userId": "${saml:sub}",
                    "saml:sub_type": "persistent"
                }
            }
        }
    ]
}
```

Replace **<Appstream Stack Resource>** with the ARN of your AppStream stack. It will be in the following format:

```
arn:aws:appstream:<your AWS region>:<your AWS account
ID>:stack/<the name of your AppStream stack>
```

Example:

```
arn:aws:appstream:us-west-2:123456789012:stack/MyFirstStack
```

Learn more about ARNs at:
http://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html

Learn more about Access Policies at:
http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

Note: For multiple appstream stacks, add another Statement object to the list with the appropriate Appstream Stack Resource.

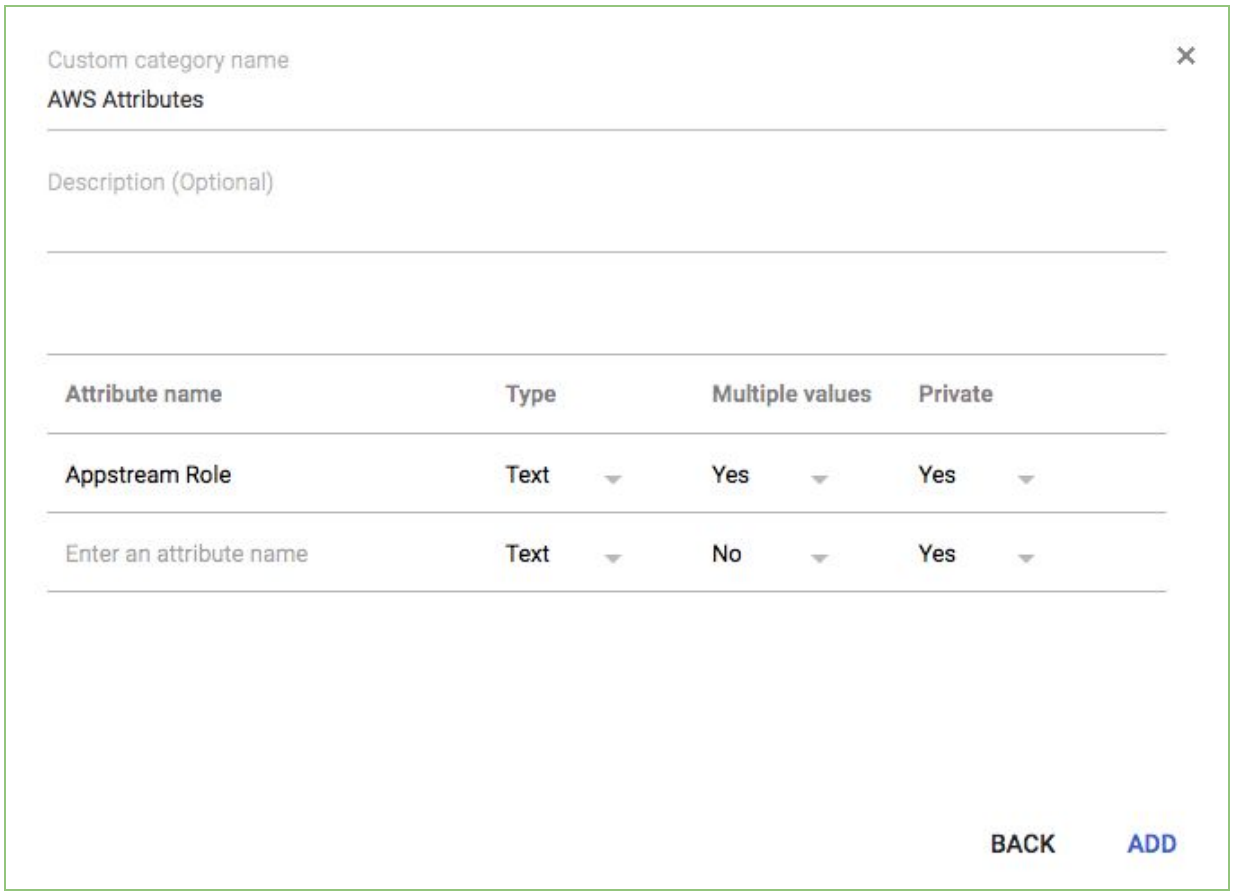
11. Click **Validate Policy**. The validation results will be displayed above “Policy Name.”
12. Once your policy is validated, click **Create Policy**.
13. Select the newly created policy from the list. Click **Next: Review**.
14. Give the role a name (e.g. GoogleAppStreamUsers) and description.
15. Click **Create role**.
16. Click the newly created role to view the summary. Note the **Role ARN**, as it will be used in a later step.

Creating custom attributes for SAML information

For this step, you will need to create a custom user attribute schema and add data to users using that schema. You can either use the Google Admin console, or the API. Both methods are explained below.

Create a custom schema (Console)

1. From the Google Admin console dashboard, go to **Users**.
2. In the toolbar, click the **Manage user attributes** button:
3. Click **Add Custom Category**.

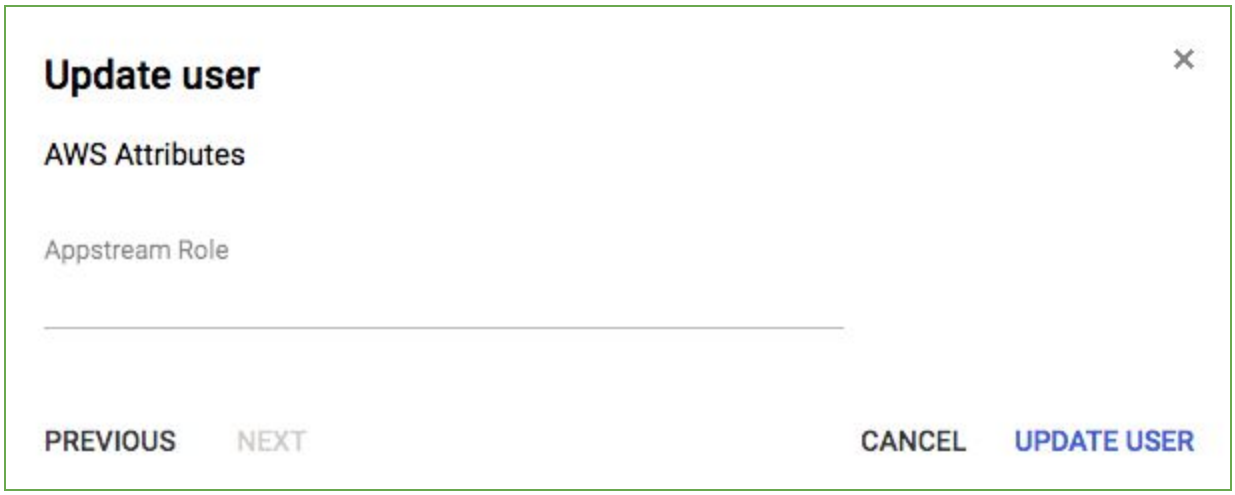


4. In the **Custom category name** field, name your new category of user attributes (e.g. “AWS Attributes”).
5. (Optional) In the **Description** field, enter a description that clarifies your new category.
6. Click **Enter an attribute name** to add a custom user attribute.
7. Create a custom attribute:
    a. **Attribute name**: Enter the label you want to display on the user’s account page (e.g. “Appstream Role”).
    b. **Attribute type**: Select “Text.”
    c. **Multiple values**: Select “Yes.”
    d. **Private**: Select “Yes.”
8. Click **Add**.

Add custom Data (Console)

1. From the Google Admin console dashboard, go to Users.
2. Click the name of a user to open their user account page.
3. Click **Account** and find the **Manage user attributes** section.

4. Click **Edit** to add custom attribute values.
    a. Find the custom category you made earlier using **Previous** and **Next** to toggle through the categories.



5. For the attribute, enter:

   <role ARN>,<provider ARN>

   Using the ARN values you noted earlier. (`<role ARN>` is at the end of “Create a Role that uses your Identity Provider.”, and `<provider ARN>` is at the end of “Create your Identity Provider.”)

   Example:
   `arn:aws:iam::0123456789012:role/GoogleAppStreamUsers,arn:aws:iam::0123456789012:saml-provider/MyGoogleProvider`

6. Click Update User.

Create a custom schema (API)

1. Open the schema insert page (all work will be done in the “Try this API” sidebar)
2. Enter "my_customer" for customerId
3. Using the request editor in the sidebar, input the following:

```
{
  "fields":
     [
       {
```

```
            "fieldName": "role",
            "fieldType": "STRING",
            "readAccessType": "ADMINS_AND_SELF",
            "multiValued": true
          }
        ],
    "schemaName": "SSO"
}
```

Notes: The schemaName and fieldName can be any text value.
If you want to use more than one role, set multiValued to true.

4. Click Execute.
   You should see a 200 OK response, and the output of the request is displayed.

Add Custom Data (API)

Now that the custom attribute exists, you need to populate that data so the SAML request can use it.

1. Open the User Update page (all work will be done in the “Try this API” sidebar)
2. Enter a valid user primary email address, alias email address, or unique user ID for userKey
3. Using the request editor in the sidebar, input the following:

```
{
  "customSchemas":
  {
  "SSO":
    {
      "role": [
        {
          "value": "<role ARN>,<provider ARN>",
          "customType": "SSO"
        }
      ]
```

```
        }
    }
}
```

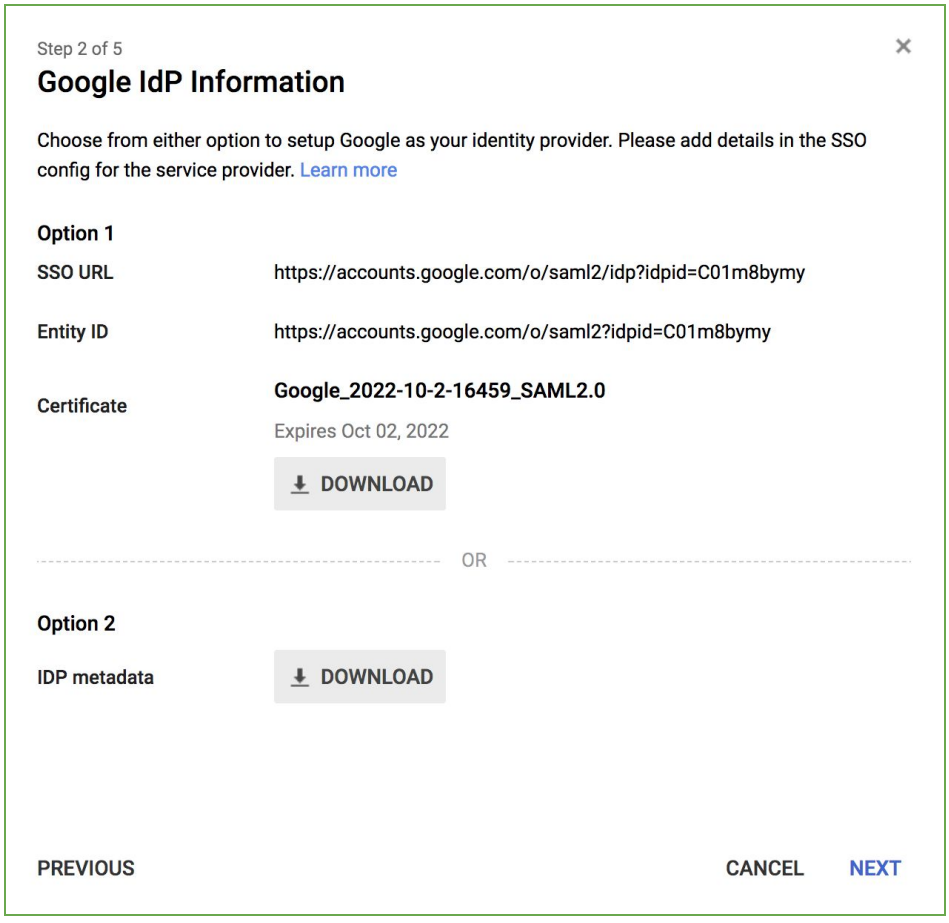
4. Click Execute.
   You should see a 200 OK response, and the user profile is updated with the custom data.

Finish creating the SAML application

Information to Find

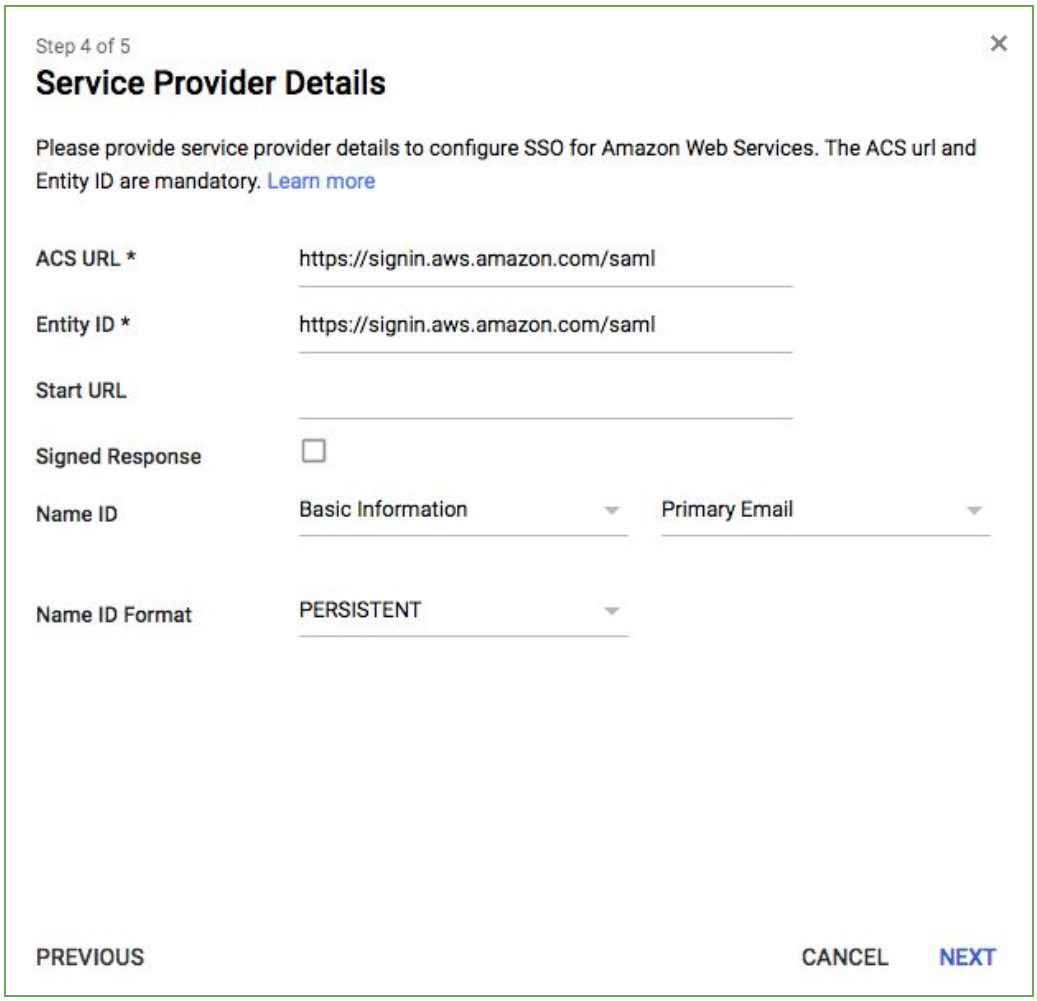
- Appstream Stack name
- AWS account ID
- Region where the Appstream Stack is located

1. Return to the Google Admin Console you left open in “Get your Google Metadata Documents.”



2. Click **NEXT**.

3. The Basic information will already be pre-filled. Click **NEXT**.



4. In the Service Provider Details, The ACS URL and Entity ID will be prefilled. Enter the **Start URL** in this format:

   `https://appstream2.<AWS region>.aws.amazon.com/saml?stack=<stack name>&accountId=<AWS account ID>`

   Example:
   `https://appstream2.us-west-2.aws.amazon.com/saml?stack=MyFirstStack&accountId=123456789012`

5. Leave “Signed Response” unchecked.

6. Leave the Name ID set to “Basic Information,” “Primary Email.”

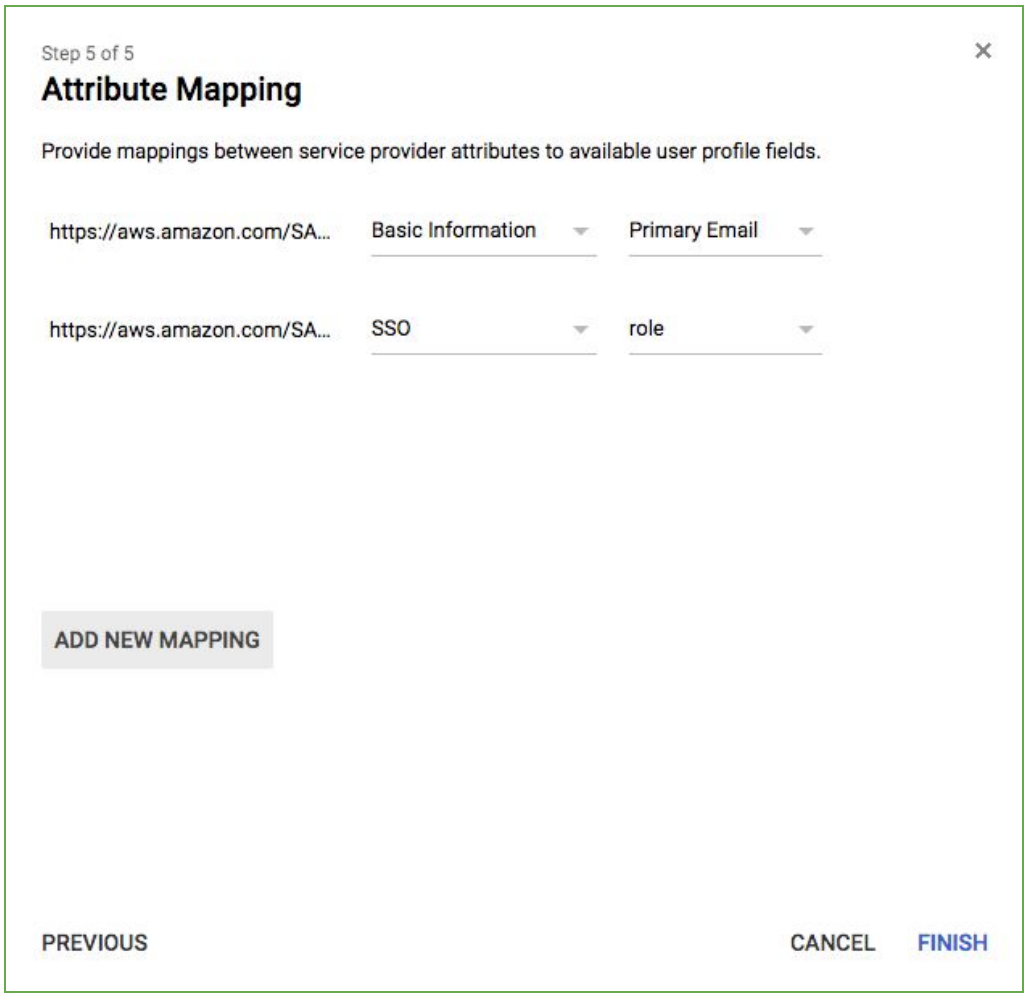
7. Set the **Name ID Format** to “PERSISTENT.”

8. Click **NEXT**.

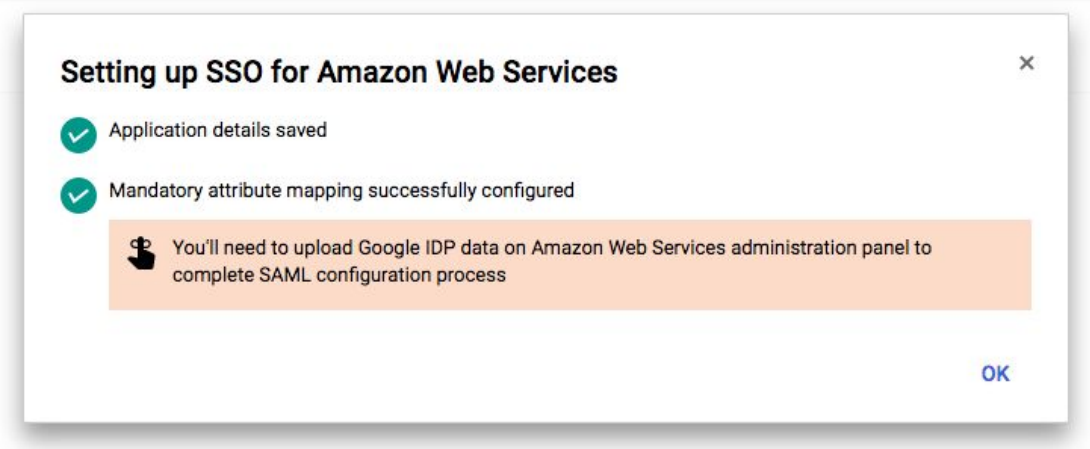
9. In Attribute mapping, you’ll use the custom attribute you created earlier. The mapping URIs may be truncated, but fill them out in the order below:

   https://aws.amazon.com/SAML/Attributes/RoleSessionName*: Choose “Basic Information,” then “Primary Email.”

   https://aws.amazon.com/SAML/Attributes/Role*: Choose the name of your custom category, then your custom attribute. (These were created in “Creating custom attributes for SAML information.”)

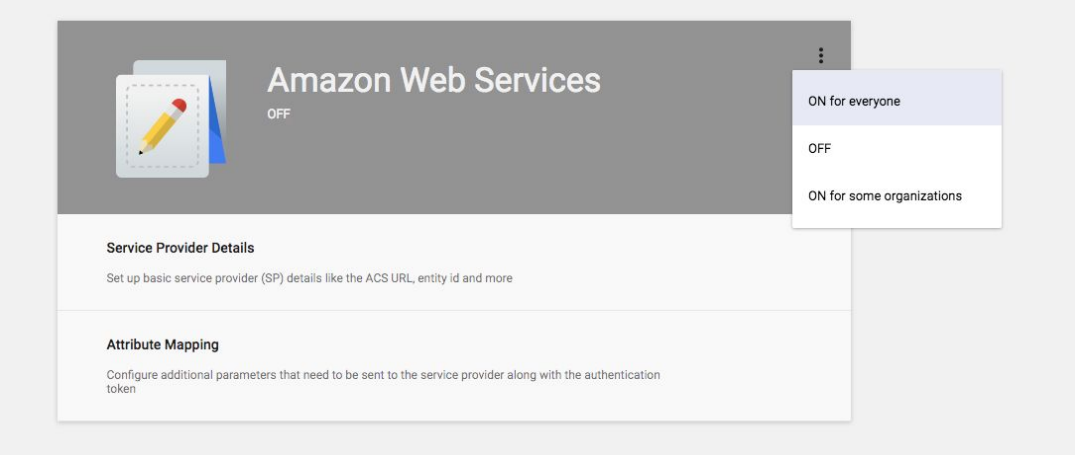

Step 5 of 5
Attribute Mapping
Provide mappings between service provider attributes to available user profile fields.
https://aws.amazon.com/SA...
Basic Information
Primary Email
https://aws.amazon.com/SA...
SSO
role
ADD NEW MAPPING
PREVIOUS
CANCEL
FINISH

10. Click **FINISH** and you should see this:



11. Close the pop-up and open the Amazon Web Services app you just made. (You may need to reload the page for it to appear.)

12. Click the options drop-down in the upper-right of the Amazon Web Services box. Choose “On for everyone” to enable for all users, or use “ON for some organizations” to enable it for specific sub-organizations.



Verify that SSO is working between Google and AWS

Note: Make sure you're signed in to an account for which you've configured user data for Amazon Web Services with these steps.

1. Open a G Suite core service, such as Google Calendar, Drive, or Gmail.
2. At the top right, click the **App Launcher**: 
3. Scroll to the apps section and click **Amazon Web Services**. (If you don't see it, your SAML app changes may not have propagated yet. They may take up to 24 hours to propagate, but in most propagate much faster than that. Wait and try again.)
    a. If you are signed in to more than one account, select the account where Amazon Web Services is configured.
    b. If you configured more than one role, select a role from the list and click Sign In.

Amazon AppStream 2.0
Chrome Packaging and Deployment

These steps will take you through creating a Chrome app and deploying it to your users as a private app. These instructions can only be completed after following the steps in “Configuring Google SSO with Amazon AppStream 2.0.”

1. Get your SAML app URL.
2. Enable app permissions for your domain.
3. Verify a website.
4. Create a Chrome hosted app.
5. Privately publish your app.
6. Force install the app.

Get your SAML app URL

Note: Make sure you're signed in to an account for which you've configured user data for Amazon Web Services.

1. In your Chrome browser, open Google Docs for your organization (https://docs.google.com).
2. At the top right, click the App Launcher: 
3. Scroll to the bottom section and find the Amazon Web Services icon.
4. Right-click the icon and choose "Copy Link Address."
5. Paste the link in a text editor for use later. It should look similar to this: https://accounts.google.com/o/saml2/initsso?idpid=XXXXXXXXX&spid=YYYYYYYYYYY&forceauthn=false

Enable app permissions for your domain

1. Open the Google Admin console.
2. Navigate to **Device management** > **Chrome management** (in left sidebar) > **User settings**.
3. Scroll down to the **Chrome Web Store Permissions** section of the user settings.
4. Check **Allow users to publish private apps that are restricted to your domain on Chrome Web Store.**
5. Check **Allow users to skip verification for websites not owned.**
6. Click **SAVE**.

Verify a website

When submitting your app, you will need to associate your app with a website that you are the verified owner of. Typically, this would be your organization's domain, but it can technically be any site.

Go to https://www.google.com/webmasters/tools/ and click **ADD A PROPERTY** to add a site. You will have several options to verify ownership.

Google's documentation on verification is at:
https://support.google.com/webmasters/answer/35179?hl=en

Create a Chrome hosted app

1. Create a folder on your local computer to contain your app source files. The folder's name should be the name of your app.
2. In the source folder, create a file named manifest.json.
3. Paste this template into manifest.json:

```
{
 "name": "<YOUR APP NAME>",
 "description": "<YOUR APP DESCRIPTION>",
 "version": "1.0",
 "manifest_version": 2,
 "icons": {
  "128": "icon_128.png"
 },
 "app": {
  "launch": {
   "web_url": "<YOUR APP URL>"
  }
 }
}
```

4. Replace **<YOUR APP NAME>** with a name (e.g. "MyDistrict AppStream"), and **<YOUR APP DESCRIPTION>** with a 1-sentence description for your app.

   Replace **<YOUR APP URL>** with the URL you saved in "Get your SAML app URL."

5. Create an app icon. It needs to be a 128×128 PNG file. Name it **icon_128.png** and add it to your source folder.
6. Test your app locally in the Chrome browser.
   a. Go to chrome://extensions.
   b. Select **Developer mode** if not already active.
   c. Click **Load unpacked extension…** and select your source folder.

d. Go to: chrome://apps. If successful, your app will appear on this page.

7. Create a .zip archive of your source folder.

Privately publish your app

1. Go to the Chrome Developer Dashboard. (https://chrome.google.com/webstore/developer/dashboard).
2. Click **Add New Item**.
3. Click **Choose File** and select the .zip archive of your app.
4. Click **Upload**. Once the upload finishes, you'll be redirected to a page to create your app listing.
5. Now you'll need to fill in some details and provide some images for your app. You will at least need to provide:
    a. **Description**
    b. **Icon**
    c. One 1280x800 or 640x400 **Screenshot**
    d. A 440x280 **Promotional tile image**
    e. A **Category**
    f. A **Language**
    g. **Websites: Verify that this is an official item for a website you own**: Choose the website you validated.
    h. **Visibility options:** Choose "Private," then "Everyone at <your domain>."
6. Click "Publish."

Force install the app

1. Open the Google Admin console.
2. Navigate to **Device management** > **Chrome management** (in left sidebar) > **User settings**.
3. Under **Filters**, choose **App Type: “**Chrome Apps” and **Type:** “Domain Apps.”
4. You should see your app in the list. Click to view it, then choose **User Settings**.
5. You’ll see an **Orgs** column with your organization and any sub-organizations. Any settings on your organization are the default for all sub-organizations, but can be overridden.
6. For each organization you want to configure for force installation:
    a. Click it in the **Orgs** column.
    b. Under **Force installation**, click **Override** if it appears under the switch, then click the switch to toggle it on.
    c. Click **SAVE**. (You may need to scroll down.)
7. The app will be installed the next time the policy updates for applicable users.