Configuring Service Workbench for authentication

Contents

Overview 2

Prerequisites 2

Service Workbench and IdP workflow 2

Advantages of configuring Service Workbench using IdP 3

Using native Amazon Cognito user pool for authentication 3

Log in using internal authentication 4

Configuring Service Workbench using IdP 4

- Configuring Service Workbench using Azure AD 4
- Configuring Service Workbench using Microsoft ADFS 5
  - Creating a Relying Party in Microsoft ADFS 5
  - Configuring Relying Party information in Service Workbench 6
  - Adding Relying Party trust for the Amazon Cognito user pool in Microsoft ADFS 7
- Configuring Service Workbench using Auth0 8
  - Configuring Auth0 8
  - Configuring SAML2 9
  - Downloading SAML2 template 12
  - Configuring Service Workbench environment 13

Overview

An identity provider (IdP) is a service that stores, manages digital identities and verifies the identities of users. The IdP can be used in place of internal authentication mechanism of Service Workbench. The IdP authenticates the identity of a user, allowing Service Workbench to assign permissions to that user.

In an IdP workflow, user login is performed by an external identify provider. Examples of these include Amazon SSO, other SAML-2.0 identity providers such as Azure Active Directory (Azure AD), or social connections. After your identity is verified by the IdP, Service Workbench assigns you the rights to access the resources for which you are authorized. For more information about IdPs, see Identity Providers for external entities.

Prerequisites

- Create a SAML 2.0-compliant metadata file using IdP. Place the metadata file here: `main/solution/post-deployment/config/saml-metadata`
- Enter the following values in your `main/config/settings/<stage.yml>`: `fedIdpIds, fedIdpNames, fedIdpDisplayNames, fedIdpMetadatas`
  Examples for these are available in `main/config/settings/example.yml`
- Deploy Service Workbench

Service Workbench and IdP workflow

While using Service Workbench, you are redirected to the configured IdP login page. Next, the IdP can authenticate using any backend authentication connection, such as an LDAP directory, or a database. The IdP data is then provided by Amazon Cognito user pool. Once you are authenticated, the IdP returns a Security Assertion Markup Language (SAML) metadata file to Service Workbench indicating that you have been successfully authenticated. It then provides user details (such as, name, email etc.) as shown below.

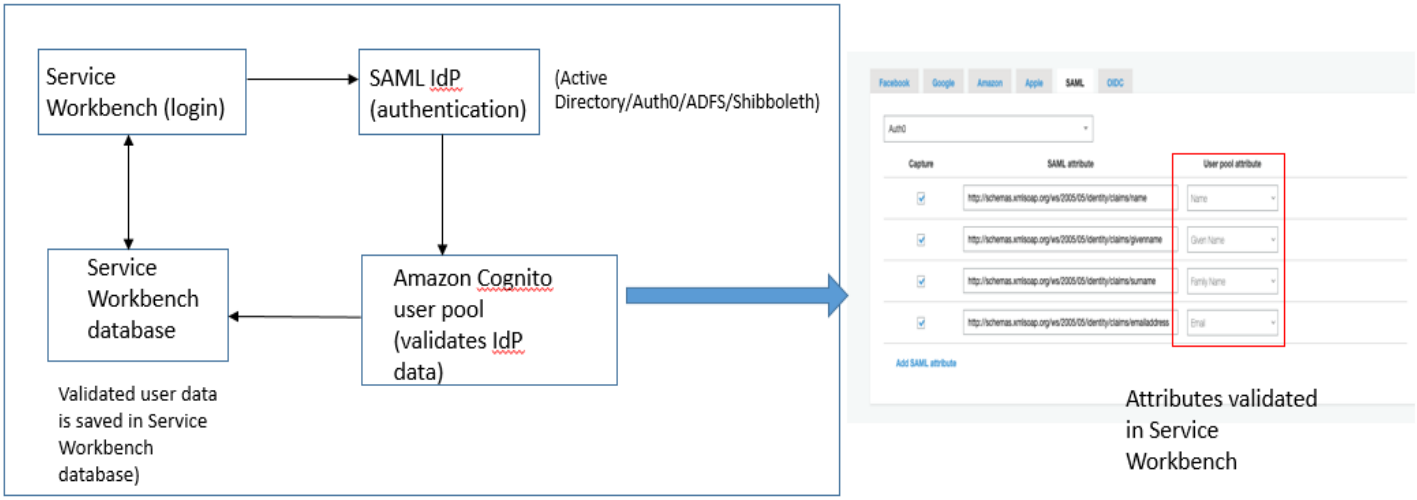


Figure 1: IdP workflow using Service Workbench

Advantages of configuring Service Workbench using IdP

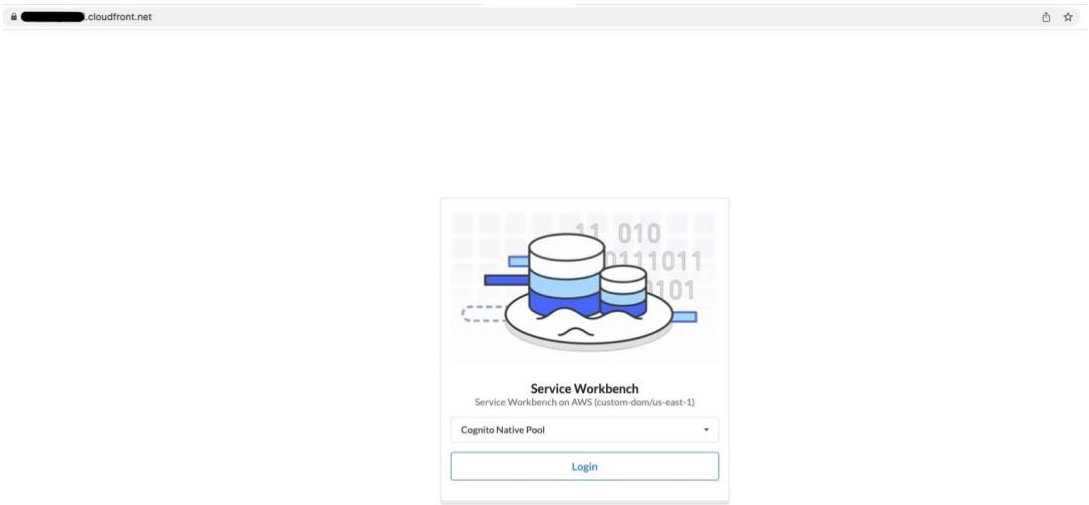
When Service Workbench is configured using an IdP, the login process is streamlined by allowing authentication federation and single sign-on (SSO) compatibility. Therefore, you can use an existing identity (and password) to log in. It also reduces the need to create custom logins and tracks user activity. Using IdP, a single login provides all data at one go.

You can access Service Workbench from a variety of devices, locations, and time zones. An IdP manages those details efficiently.

Using native Amazon Cognito user pool for authentication

Note: As a security enhancement, the internal authentication method used by Service Workbench (the legacy default authentication method) will soon be deprecated.

In Service Workbench 4.2.0, the native Amazon Cognito user pool is the default authentication method, and is reflected accordingly on the application's login page (alongside your external SAML IdP integrations, if any).



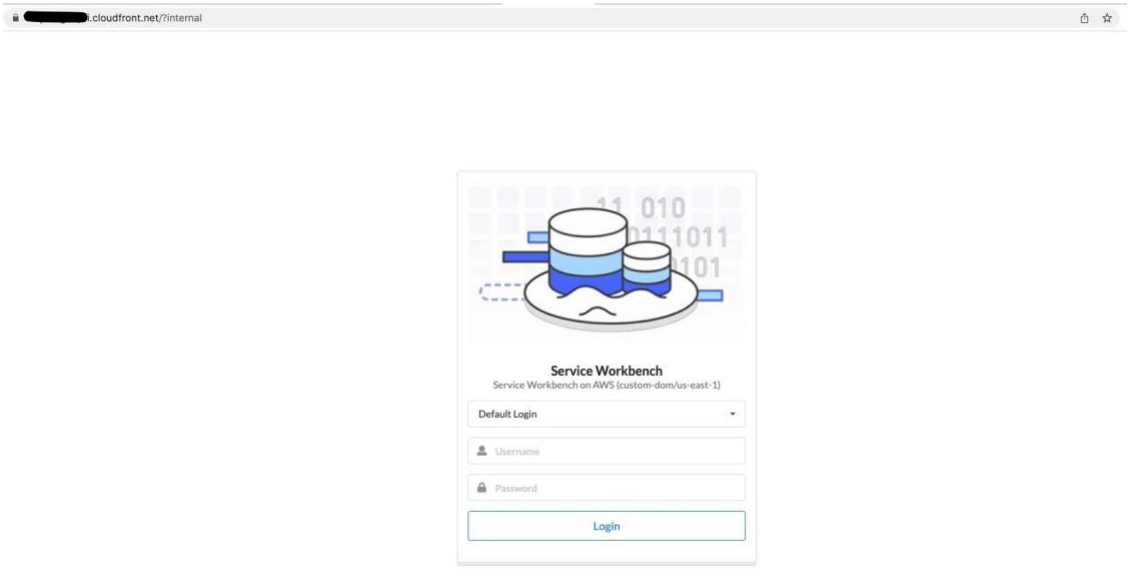
You will find the default (user-customizable) configurations determining the native Amazon Cognito user pool behavior in the `main/solution/post-deployment/config/settings/.defaults.yml` file.

By default, you can sign up onto the native pool, but can only access Service Workbench once you are approved by the Service Workbench administrator. The user addition experience on Service Workbench for native Amazon Cognito user pool is similar to that of an external IdP.

A new administrator user is created in Service Workbench using the `rootUserEmail` value as provided by your stage configuration. A temporary password is available in the installation summary necessary for logging in the native administrator user for the first time.

Log in using internal authentication

You can still log in to Service Workbench using the internal authentication method by adding `/?internal` to your Service Workbench URL (for example, `https://<random_string>.cloudfront.net/?internal`).



Important: We suggest creating new users in native Cognito user pool (or an external IdP, if you use one) corresponding to their internal authentication counterparts, and migrating resource permissions over to these new users.

Configuring Service Workbench using IdP

The following topics provide information about configuring Service Workbench by using the IdPs (examples below):

- Configuring Service Workbench using Azure AD
- Configuring Service Workbench using Microsoft ADFS
- Configuring Service Workbench using Auth0

Configuring Service Workbench using Azure AD

You can use Azure AD to create and manage domains, users, and objects within a network. It provides a way to organize a large number of users into logical groups and subgroups. It also provides access control at each level. For more information about IdPs, see Identity Providers.

Azure AD (or any IdP) is a source of authentication. It authenticates users for Service Workbench login. After successful Azure AD login, it sends user information to an Amazon Cognito user pool created by

Service Workbench. Service Workbench then uses the Amazon Cognito user pool for its internal use as described in Using Service Workbench with IdP.

To configure Azure AD authentication:

1. Create an IdP if you don't have one. For more information about creating an IdP, see sign up your organization.
2. Download SAML metadata (XML file).
3. Using Amazon Cognito on the AWS Management Console, create an Amazon Cognito user pool. The name of the pool must be <*stage*>-<*solution_name*>-userPool, where *stage* and *solution_name* are configured in the main configuration file.
4. Gather the relying party information, such as **User Pool Id**, **Relying Party Id**, and **User Pool Signing Cert**.
5. Run the following script from the root of Service Workbench repository:
   `scripts/get-relying-party.sh`
6. Copy the output of this script and provide it to your Azure AD administrator.

```
------------------------------------------------------------------------
Summary:
------------------------------------------------------------------------


User Pool Id                                : <id of the AWS Cognito User Pool>
Relying Party Id (Cognito User Pool URN)    : <URN of the AWS Cognito User Pool that is used as relying party id (RPID)>
(Login) SAML Assertion Consumer Endpoint    : <URL of the SAML assertion consumer endpoint>
(Logout) SAML Logout Endpoint               : <URL of the SAML logout endpoint>
User Pool Signing Cert                      : <Signing certificate from the AWS Cognito User Pool>
Solution                                    : <The name of the solution as specified in the settings configuration file>
Environment Name                            : <Name of the environment (i.e., stage) as specified in the settings configuration file>
```

Figure 2: Script output

Configuring Service Workbench using Microsoft ADFS

Microsoft ADFS is another IdP and it uses SAML 2.0 standard. For SAML federation, the Amazon Cognito user pool is the Service Provider (SP). Mutual trust between the SP and the IdP must be established. Service Workbench on AWS uses Amazon Cognito user pools to federate identities from Microsoft Active Directory using ADFS and SAML 2.0.

Creating a Relying Party in Microsoft ADFS

Follow these steps to establish trust from the IdP (Microsoft ADFS) to Service Provider (Amazon Cognito user pool) and create a relying party in Microsoft ADFS. See Using Service Workbench with IdP.

Establishing trust between Microsoft ADFS (IdP) and Amazon Cognito user pool

1. Log in to your Active Directory domain controller. Type `AD FS` in the **Run** window and open **AD FS 2.0 Management**.
2. Choose **Add Relying Party Trust**.
3. Choose **Start** on the **Welcome** screen.
4. For **Select Data Source**, choose **Enter Data About the Relying Party Manually**.
5. For **Specify Display Name** pane, enter a display name and relevant notes about the relying party. For example, enter **Amazon Cognito User Pool Relying Party**.
6. For **Choose Profile**, choose **AD FS 2.0 Profile**.

7. Do not configure any certificate in the **Configure Certificate** pane. Configuring certificates is for encrypting SAML claims. The SP (that is, the Amazon Cognito user pool) needs a private key to decrypt the claims if you configure a certificate. The Amazon Cognito user pool does not currently support encrypted SAML assertions.
8. Do not select any option in the **Configure URL** pane. Choose **Next**.
9. At this point, there is no need to configure anything in the **Configure Identifiers** pane.
10. In the **Choose Issuance Authorization Rules** pane, choose **Permit All Users to Access this Relying Party**.
11. For **Ready to Add Trust**, choose **Next**.
12. For **Finish**, choose **Close.**

Configuring IdP attributes

1. Configure the attributes that you want for the SAML assertion. The attributes are read by the Amazon Cognito user pool and aligned to the standard Amazon Cognito attributes from the mapping configuration in Amazon Cognito.
2. The **Edit Claims** window may already be open from the last wizard. If not, you can open it by choosing the **Edit Claim Rules** link and configure the claims. Add the following claims:

    - **Name ID** (Optional)
    - **Name**
    - **Email**
    - **Surname**
    - **Given Name**

3. For **Name ID** claim, choose **Add Rule**.
4. Choose **Transform an Incoming Claim**, then choose **Next** and configure the claim.
5. Follow the same actions from to add the **Name** claim.
6. To add the **Email** claim, choose **Add Rule**.
7. For **Send LDAP Attributes as Claims**, choose **Next** and then configure the claim.
8. Similarly, add the **Surname** and **Given Name** claims.

Configuring Relying Party information in Service Workbench

After creating a relying party in Microsoft ADFS, you can configure it within the Service Workbench. Follow these steps to configure the relying party.

1. Extract the SAML metadata file from Microsoft ADFS. The location of the metadata file might be different depending upon your version of Microsoft AD/ADFS. By default, it is located at:

    ```
    https://<DomainControllerDNSName>/FederationMetadata/2007-
    06/FederationMetadata.xml
    ```

2. Copy the above metadata file and place it at the following location:

    ```
    /solution/post-deployment/config/saml-metadata/metadata.xml
    ```

3. Modify the component-specific settings file for post-deployment at the following location:

```
/solution/post-deployment/config/settings/<your-environment-name>.yml
```

4. Enter the `fedIdpMetadatas` settings.

```
fedIdpMetadatas:'["s3://${self:custom.settings.namespace}-
artifacts/saml-metadata/metadata.xml"]'
```

Adding Relying Party trust for the Amazon Cognito user pool in Microsoft ADFS

After you have deployed the solution, an Amazon Cognito user pool is created. Follow these steps to add a relying party trust for the Amazon Cognito User Pool:

1. Sign in to the AWS Management Console and navigate to Amazon Cognito.
2. Choose **User Pool** to see a user pool for your environment. The Amazon Cognito user pool is specified in the following format:

   `<envName>-<solutionName>-userpool`

   where, <*envName*> and <*solutionName*> denote the values specified in your settings file.

3. Choose the user pool for your environment and note the following values:

   - **User Pool ID**: Copy the value for field **Pool ID**.
   - **Domain Prefix**: Navigate to **App Integration domain Name** for your **User Pool** and copy the value of the **domain Prefix**.

4. Log in to Microsoft ADFS domain controller and add the Amazon Cognito user pool-related information.

   a) Open the Microsoft ADFS Management application.
   b) Navigate to **Relying Party Trusts**.
   c) To add trust, choose the appropriate **Relying Party**.
   d) Open the **Identifiers** tab.
   e) Enter the URN of the Amazon Cognito User Pool and choose **Add**.

5. Replace the `<userPoolId>` with the value of the **User Pool ID** you obtained earlier. The URN is in the following format:

   `urn:amazon:cognito:sp:<userPoolId>`

6. Open the **Endpoints** tab and add the SP URL that receives the SAML assertion from the IdP. The SP is the consumer of the Amazon Cognito SAML assertion.
7. Replace `<userPoolDomain>` with the value of **domain Prefix**.
8. Replace `<region>` with the region in which you deployed the solution. The URL is in the following format:

```
https://<userPoolDomain>.auth.<region>.amazoncognito.com/saml2/idpresponse
```

You should now be able to log in to Service Workbench using Microsoft Active Directory credentials.

Configuring Service Workbench using Auth0

You can configure Service Workbench using Auth0 by first creating an application in Auth0. Next, configure SAML and download the SAML template. Finally, configure the Service Workbench environment.

Configuring Auth0

Auth0 is another IdP, which used for adding authentication and authorization to your applications. Service Workbench on AWS can be configured to authenticate users through Auth0 and SAML2. For more information about using Auth0, see Test SAML SSO with Auth0 as Service Provider and Identity Provider.

Prerequisites

You must have an existing account with Auth0.

Creating an application using Auth0

To create an application:

1. Log in to your Auth0 account and navigate to the **Applications** page.
2. If an application does not already exist, choose the **Create Application** button.

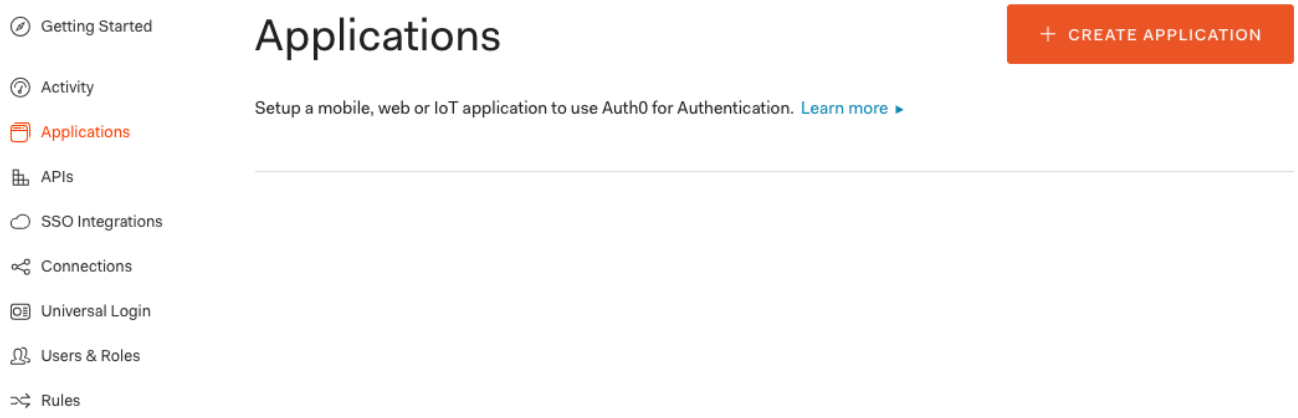


Figure 3: Applications webpage of Auth0.com

3. Select the application type as **Single Page Web Applications** and choose **Create**.

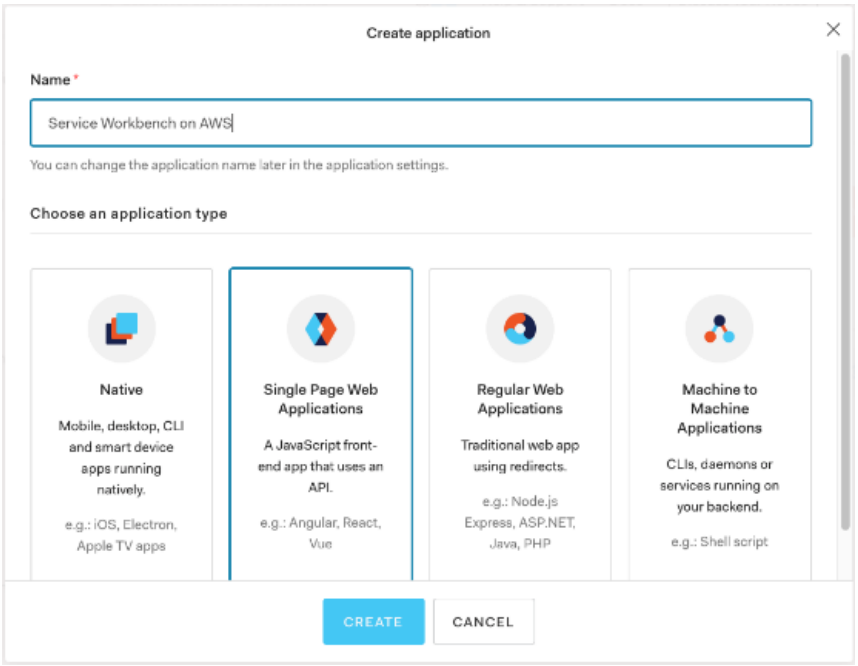


Figure 4: Application types

Configuring SAML2

SAML is an XML-based open standard for transferring identity data between two parties: an identity provider (IdP) and a service provider (SP). For more information about SAML authentication, see [How does SAML Authentication Work?]

To configure SAML2:

1. Navigate to the **Addons** tab and enable **SAML2 WEB APP**.

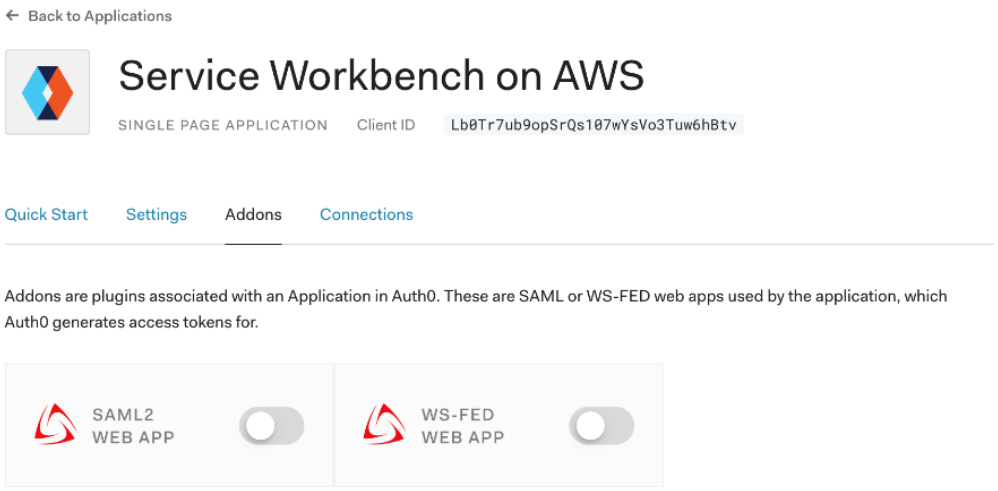


Figure 5: Addons tab for Single Page Web Applications

2. In the A**pplication Callback URL** field, enter the following URL. Replace *STAGE_NAME* and *SOLUTION_NAME* with the values from the Service Workbench settings file and *REGION* with the appropriate region.

   ```
   https://`STAGE_NAME-
   SOLUTION_NAME`.auth.REGION.amazoncognito.com/saml2/idpresponse
   ```

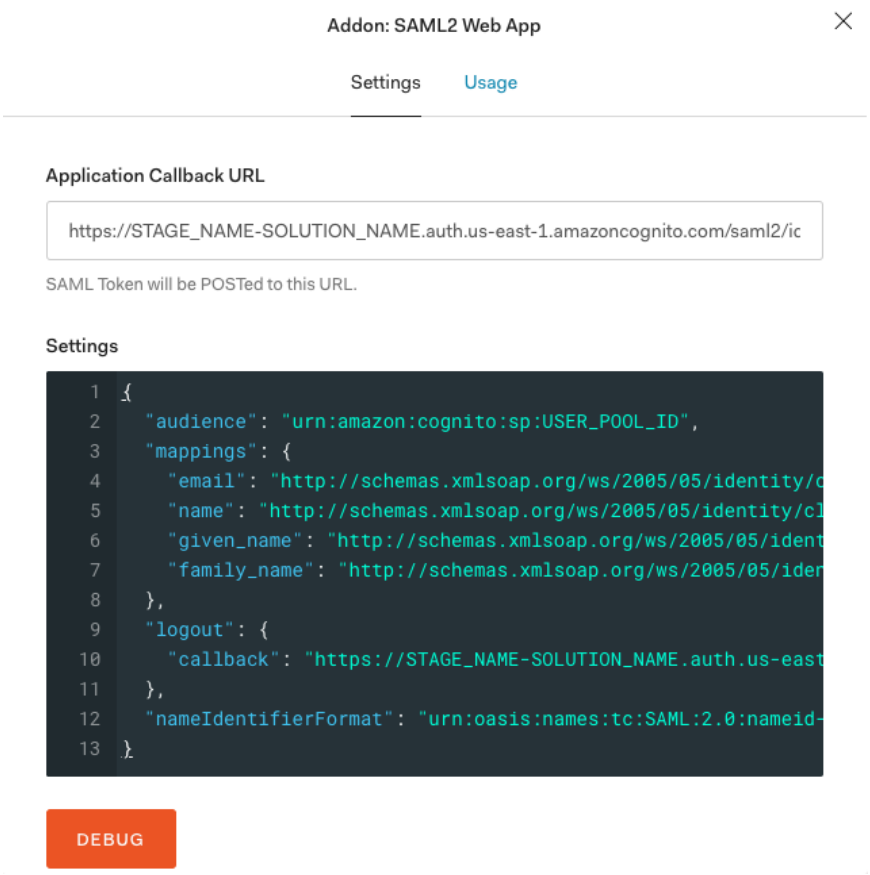


Figure 6: Application callback URL page and settings

3. Enter the JSON code into the settings block and replace <*USER_POOL_ID*> with the Service Workbench Amazon Cognito user pool ID value found in the Amazon Cognito console.
4. Enter same values for the <*STAGE_NAME*>, <*SOLUTION_NAME*>, and <*REGION*> as in the previous step.
5. After entering JSON with the appropriate values, scroll to the bottom and choose **Save**.

```
{

"audience": "urn:amazon:cognito:sp:USER_POOL_ID",

"Mappings": {

"Email":
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",

"name": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name",

"given_name":
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname",
```

```
"family_name":"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname"

},

"logout": {

"callback":"https://STAGE_NAME-SOLUTION_NAME.auth.REGION.amazoncognito.com/saml2/logout"

},

"nameIdentifierFormat":"urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

}
```

For more information about IdP workflow, see Using Service Workbench with IdP.

Downloading SAML2 template

To download SAML metadata:

1. Choose **SAML2 Web App** again, and go to the **Usage** tab.
2. Choose **Download** to download the SAML metadata XML file locally.

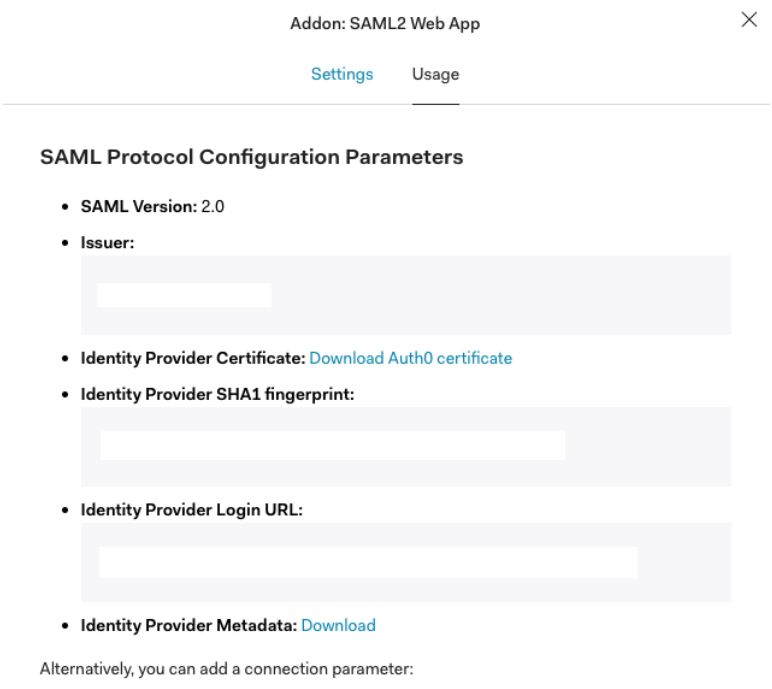


Figure 7: Usage tab of the SAML2 web application

3. Rename and place the downloaded file in the repository at the following location:

   `main/solution/post-demployment/config/saml-metadata/auth0_com-metadata.xml`

Configuring Service Workbench environment

Add the following items to the `$STAGE.yml` settings file for the environment (replace `DOMAIN` with your Auth0 domain.

```
fedIdpIds: '["Domain"]'

fedIdpNames: '["Auth0"]'

fedIdpDisplayNames: '["Auth0"]'

fedIdpMetadatas:'["s3://${self:custom.settings.deploymentBucketName}/saml-metadata/auth0_com-metadata.xml"]'
```

Finally, redeploy the system using the `scripts/environment-deploy.sh` *`STAGE_NAME`* command.

The reference documentation can be found here.

References:

- Integrating Third-Party SAML Identity Providers
- Adding SAML IdPs to a user pool