Pivotal

PRODUCT DOCUMENTATION

Single Sign-On for PCF®

Documentation

Version 1.4

Published: 11 March 2019

Table of Contents

Single Sign-On Overview

Note: Single Sign-On for PCF v1.4 is no longer supported. The support period for v1.4 has expired. To stay up-to-date with the latest software and security updates, upgrade to a supported version.

This topic provides an overview of the Single Sign-On service for Pivotal Cloud Foundry (PCF).

The Single Sign-On service is an all-in-one solution for securing access to applications and APIs on PCF. The Single Sign-On service provides support for native authentication, federated single sign-on, and authorization. Operators can configure native authentication and federated single sign-on, for example SAML, to verify the identities of application users. After authentication, the Single Sign-On service uses OAuth 2.0 to secure resources or APIs.

Single Sign-On

The Single Sign-On service allows users to log in through a single sign-on service and access other applications that are hosted or protected by the service. This improves security and productivity since users do not have to log in to individual applications.

Developers are responsible for selecting the authentication method for application users. They can select native authentication provided by the User Account and Authentication (UAA) or external identity providers. UAA is an open source identity server project under the Cloud Foundry (CF) foundation that provides identity based security for applications and APIs.

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

OAuth 2.0 Authorization

After authentication, the Single Sign-On service uses OAuth 2.0 for authorization. OAuth 2.0 is an authorization framework that delegates access to applications to access resources on behalf of a resource owner.

Developers define resources required by an application bound to a Single Sign-On (SSO) service instance and administrators grant resource permissions. See the Configure Applications topic for more details.

Product Snapshot

The following table provides version and version-support information about Single Sign-On for PCF:

Element	Details
Version	v1.4.6
Release date	November 9, 2017
Compatible Ops Manager version(s)	v1.11 or later
Compatible Elastic Runtime version(s)	v1.11 or later
IaaS support	AWS, GCP, OpenStack, and vSphere

Upgrading to the Latest Version

Consider the following compatibility information before upgrading Single Sign-On for PCF. Pivotal recommends upgrading PCF before upgrading SSO to the supported version. For example, when upgrading from PCF v1.10 to PCF v1.11, upgrade PCF so that SSO v1.3 is running on PCF v1.11, and then upgrade SSO v1.3 to SSO v1.4 as soon as possible.

Elastic Runtime Version	Supported Upgrades from SSO Versions	
	From	To
1.6.x	1.0.1–1.0.25	1.0.26
1.7.x	1.0.1–1.0.26	1.1.4
	1.1.0–1.1.3	
	1.1.0–1.1.4	

1.8.x	1.2.0–1.2.3	1.2.4
1.9.x & 1.10.x	1.2.0–1.2.4	1.3.6
1.11.x	1.3.0-1.3.6	1.4.x
1.12.x	1.4.1-1.4.6	1.5.3
	1.5.0-1.5.2	
2.0.x	1.5.3	1.6.0
2.1.x	1.6.0	1.6.x

Note: The Single Sign-On service tile operates in lockstep with Elastic Runtime.

- The SSO v1.1.x tiles are compatible with PCF v1.7.x
- The SSO v1.2.x tiles are compatible with PCF v1.8.x and later
- The SSO v1.3.x tiles are compatible with PCF v1.9.x and later
- The SSO v1.4.x tiles are compatible with PCF v1.11.x and later

Note: SSO v1.4.1–v1.4.3 are not compatible with PCF v1.12 without using the workaround in the corresponding Knowledge Base article.

Single Sign-On for PCF

- Installation
- Getting Started with Single Sign-On
- Using the System Plan
- Manage Service Plans
- Manage Service Instances
- Configure Identity Providers
- Identity Provider Discovery
- Manage Users
- Configure Applications
  - Authorization Code Grant Type
  - Implicit Grant Type
  - Client Credentials Grant Type
  - Resource Owner Password Credentials Grant Type
- Manage Resources

Active Directory Federation Services (AD FS) Integration Guide

- Active Directory Federation Services Integration Guide
  - Configure Active Directory Federation Services as an Identity Provider
  - Configure SSO Service
  - Testing
  - Troubleshooting

Azure Active Directory Integration Guide

- Azure Active Directory SAML Integration Guide
  - Configure Azure Active Directory as a SAML Identity Provider
  - Configure SSO Service
  - Testing
  - Troubleshooting

CA Single Sign-On Integration Guide

Google Cloud Platform OpenID Connect Integration Guide

Okta Integration Guide

PingFederate Integration Guide

PingOne Cloud Integration Guide

Additional Information

Release Notes

View Release Notes for Another Version

To view the release notes for another product version, select the version from the drop-down list at the top of this page.

v1.4.x

v1.4.6

Release Date: November 9, 2017

- PCF updated stemcell to 3445 series. This is a security upgrade to bump Ubuntu stemcells for USN-3420-2: Linux kernel (Xenial HWE) vulnerabilities.

v1.4.5

Release Date: October 24, 2017

- This release addresses an issue with managing service instances when more than 50 Space Developers exist within a space.

v1.4.4

Release Date: September 27, 2017

- This release addresses the upgrade issues for the Single Sign-On Service tile when legal footer links are configured.
- This release addresses a Java Buildpack issue that causes required memory to increase to 1GB.

v1.4.3

Release Date: August 30, 2017

- This is a security upgrade that resolves the following CVEs:
  - CVE-2017-8040
  - CVE-2017-8041
  - CVE-2017-8044

Additional information can be found at https://pivotal.io/security.

v1.4.2

Release Date: June 21, 2017

- PCF updated stemcell to 3363 series. This is a security upgrade to bump Ubuntu stemcells for USN-3334-1: Linux kernel (Xenial HWE) vulnerabilities.

v1.4.1

Release Date: June 15, 2017

What's New

- Application bootstrapping is available for app developers to automates and quickly onboard client and resource configurations to Single Sign-On. See

the Set Up PCF Apps to Use SSO topic for more information.

- Admin User Management interface enables plan administrators to browse and manage users across their identity providers through a simple user interface as opposed to through the UIs to help improve productivity. See the Manage Users topic for more information.
- OpenID Connect and LDAP identity provider interfaces are now available. Plan administrators can now improve their productivity by utilizing a simple user interface to configure and maintain their identity providers. See the Configure Identity Providers topic for more information.
- You can allow custom attributes to be made available through the /userinfo endpoint for your external identity providers. This can streamline user attributes for your applications. Your application must have the `user_attributes` token once custom attributes mappings and `Persist Custom Attributes` are configured for your identity provider.
- You can configure required user groups through bootstrapping. Require user groups are groups that users must have in order to authenticate to your application and obtain an access token.

Installation

This topic explains how to install Single Sign-On (SSO) for Pivotal Cloud Foundry.

Prerequisites

- Pivotal Cloud Foundry (Ops Manager and Elastic Runtime) version 1.11 or later.
- SSL Certificates.
- Application Security Groups.

Install SSO via Ops Manager

1. From Pivotal Network, select a **Single Sign-On** tile version and download the product release file.
2. From the Ops Manager Installation Dashboard, select the **Import a Product** button to upload the product file.
3. Click the plus sign icon next to the uploaded product to add this product to your staging area.
4. Click on the **Single Sign-On** tile to enter any configurations.

   Note: The Single Sign-On service tile requires a network with only one subnet until version 1.3.0. Starting with 1.3.1 multiple subnets are supported.

   Note: The SSO Identity Service Broker is deployed as a PCF application from a BOSH errand, and has no associated BOSH VMs that require selecting a corresponding network. If you are forced to select a network during installation, select the **Deployment** network, also known as the PAS or ERT network.

5. Click **Apply Changes** to install the product.

Update SSL and Load Balancer

You must update the SSL certificate for the domains listed below for each plan you create. Depending on your infrastructure and load balancer, you must also update your load balancer configuration for the following domains:

- `*.SYSTEM-DOMAIN`
- `*.APPS-DOMAIN`
- `*.login.SYSTEM-DOMAIN`
- `*.uaa.SYSTEM-DOMAIN`

Configure Application Security Groups

The Single Sign-On service requires the following network connections:

- TCP connection to load balancer(s) on port 443
- TCP and UDP connection to Domain Name Servers on port 53
- (Optional) TCP connection to your external identity provider on port 80 or 443

To enable access to the Single Sign-On service, you must ensure your Application Security Group allows access to the load balancer(s) and domain name servers that provide access to Cloud Controller and UAA. Optionally, you can configure access to your external identity provider to receive SAML metadata. For more details on how to set up application security groups, see the Application Security Groups topic.

Getting Started with Single Sign-On

This topic outlines the steps for installing and configuring the Single Sign-On service.

Install and Set Up SSO for Applications

1. Install Single Sign-On via Ops Manager.
2. Create a service plan. The Single Sign-On service is a multi-tenant service, and a service plan corresponds to a tenant. This allows an enterprise to segregate users or environments using plans. Each service plan is accessible at a tenant-specific URL in the format `https://AUTH-DOMAIN.login.SYSTEM-DOMAIN`.
3. Create a service instance. Single Sign-On service plans can provide single sign-on capabilities for applications in various spaces. A service instance lets you bind an application to a service plan.
4. Configure an identity provider. In addition to the Internal User Store, you can configure external identity providers to provide single sign-on to applications.
5. Configure your applications. Single Sign-On supports both Pivotal Cloud Foundry-hosted applications as well as externally hosted applications. Your applications must be able to request an OAuth or OpenID Connect token.
6. Create resources for your applications. If your registered applications need to make external API calls, you can assign the API endpoints as resources permitted for the application. This will whitelist the endpoints for use by the application or client.

SSO User Roles

A user's role determines which parts of an SSO configuration it can manage. SSO uses the existing user roles PCF Administrator and Space Developer, as well as a SSO-specific Plan Administrator role. This chart shows the management permissions for each role.

Management access by role	PCF Administrator	Plan Administrator	Space Developer
Service plans	X		
Service instances	X	X	X
Identity providers	X	X	
Applications	X	X	X
Resources	X	X	X

Using SSO for Pivotal Cloud Foundry Components

In addition to applications, SSO supports single sign-on for components of Pivotal Cloud Foundry, including Ops Manager and Apps Manager. This allows users already managed in an external identity provider to sign into Pivotal services. Refer to the following pages for instructions on configuring SSO to enable users in an external identity store to access PCF components:

- Ops Manager, on Amazon Web Services, vSphere, or OpenStack
- Apps Manager

Using the System Plan

This topic explains how to use the system plan for the Single Sign-On (SSO) service for Pivotal Cloud Foundry (PCF). The system plan is the default plan meant for developer apps, not end-user apps.

SSO for PCF comes with a default `system` plan that has the following features:

- Read-only
- Minimal configuration options
- Not deletable
- Allows developer-level access to system components like Elastic Runtime and its APIs
- Available only to PCF administrators

Restricting the visibility of this system plan to a single, developer-apps only org secures system components, following the principle of least privilege.

Examples of developer apps include scripts or pipelines that push other apps and services. Any app that uses the Cloud Foundry API is a developer app.

System Plan Best Practice

Pivotal recommends configuring your orgs and SSO plans as follows to prevent anyone from applying the system plan to end-user apps:

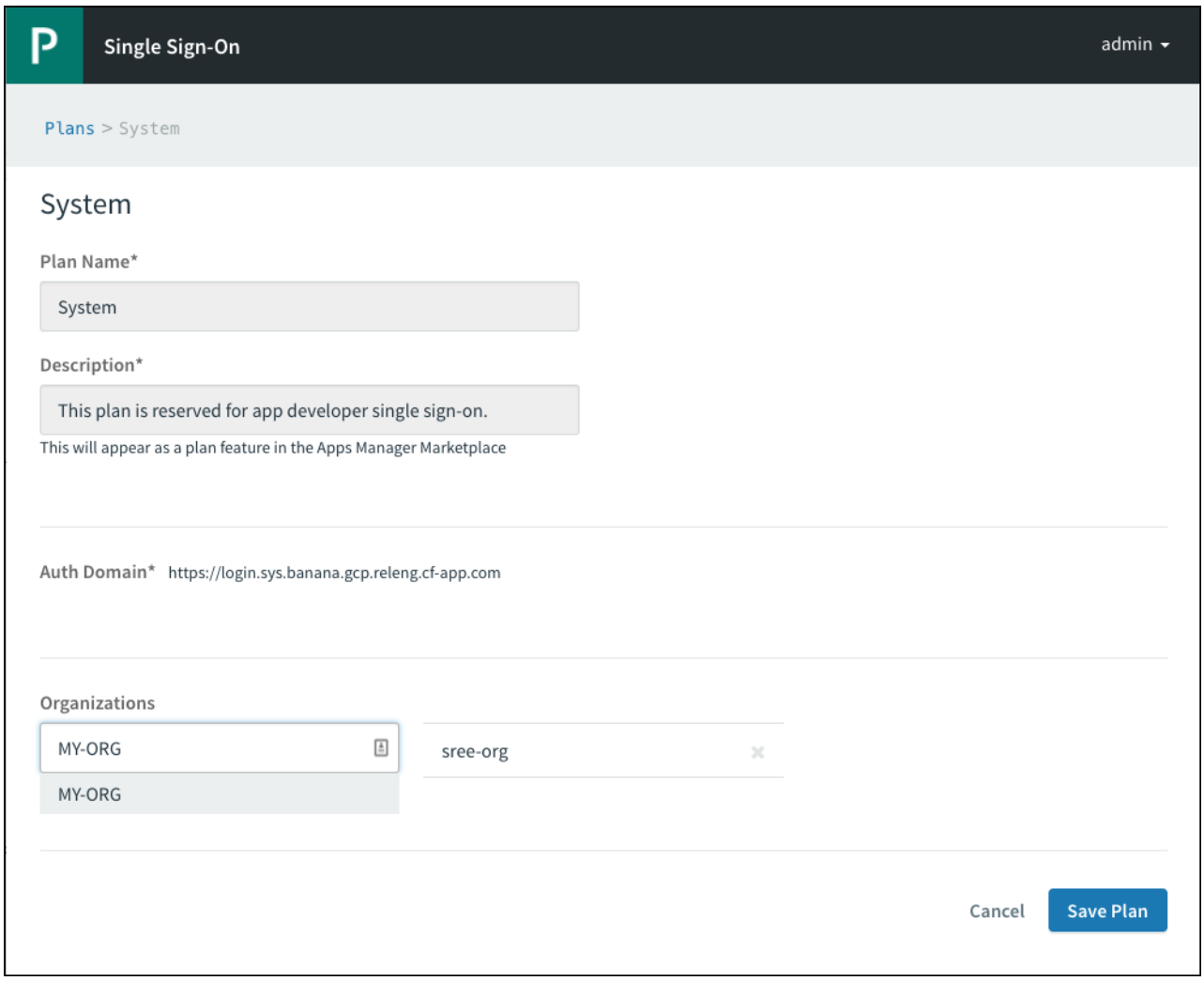
1. Restrict all developer apps to a single org.
2. Make the system plan visible only to the developer-apps org.
3. Configure other orgs with SSO service plans of their own.

Developers can then self-register their developers apps in the developer-apps org for use by other developers.

Administrators: Configure the System Plan for an Org

PCF administrators follow the steps below to enable the system plan and provide access to app developers:

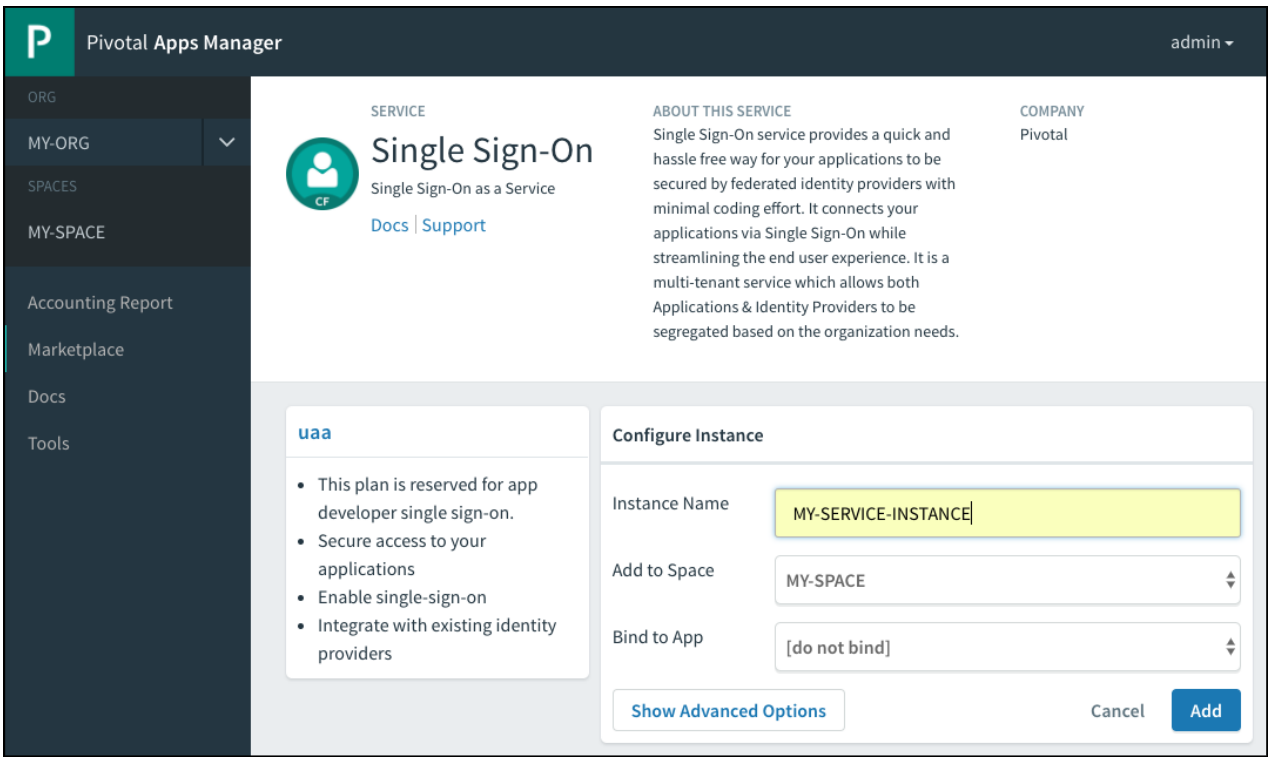
1. Log into the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. In your Elastic Runtime tile in Ops Manager, the **Domain** settings show your system domain, and the **Credentials** tab shows the **UAA Admin Credentials**.
2. Navigate to the System Plan and enable the plan in the relevant org(s).



Developers: Create a System Plan Instance for your App

Follow the steps below to create and use the `system` service plan with your developer apps.

1. Follow the steps to Create a Service Instance of SSO.



2. If your app runs on PCF, bind the application with the service instance you created. See the Bind an Application Hosted on PCF topic for more information.

3. If your app is a pipeline or a script that runs external to PCF but calls PCF APIs, do the following:

    a. Follow the instructions to Register an External Application and use the guidelines below:

        - Choose **Native App** for your application type.
        - In the app configuration, set a value for the **Refresh token lifetime** based on your use case for automated access.

    b. To give your your pipeline or script access to your resources without your presence, embed a refresh token instead of hardcoding your credentials:

        i. Run `uaac token sso get`.
        ii. At the prompts, enter the Client ID and Secret from the **Next Steps** section of the SSO dashboard. Copy the authentication URL from the command output.
        iii. Paste the authentication URL into a browser, and log in using your UAA Admin Credentials.
        iv. Copy the **Temporary Authentication Code** from the browser into the UAAC to finish the authentication.
        v. Run `uaac context`.
        vi. Copy the value of the refresh token and use that in your code to get a new token based on your client id and secret using the standard OAuth refresh token flow as described in the UAA API documentation.

Developers: Revoke System Plan Access for an External App

To revoke system plan access from an app that is external to PCF and is registered with the system plan to access PCF components, do one of the following:

- Regenerate the App Secret
- Delete the app

Manage Service Plans

This topic describes how Pivotal Cloud Foundry (PCF) Administrators manage Single Sign-On service plans.

Single Sign-On is a multi-tenant service, which enables a deployment to host multiple tenants as service plans. Each service plan can have its own administrators, applications and users. This lets enterprises segregate access by using separate plans. For example, the following tenants might require separate plans:

- Business units and geographical locations
- Employees, consumers, and partners
- Development, staging, and production instances

Administrators can create new Single Sign-On service plans at any time from the SSO dashboard.

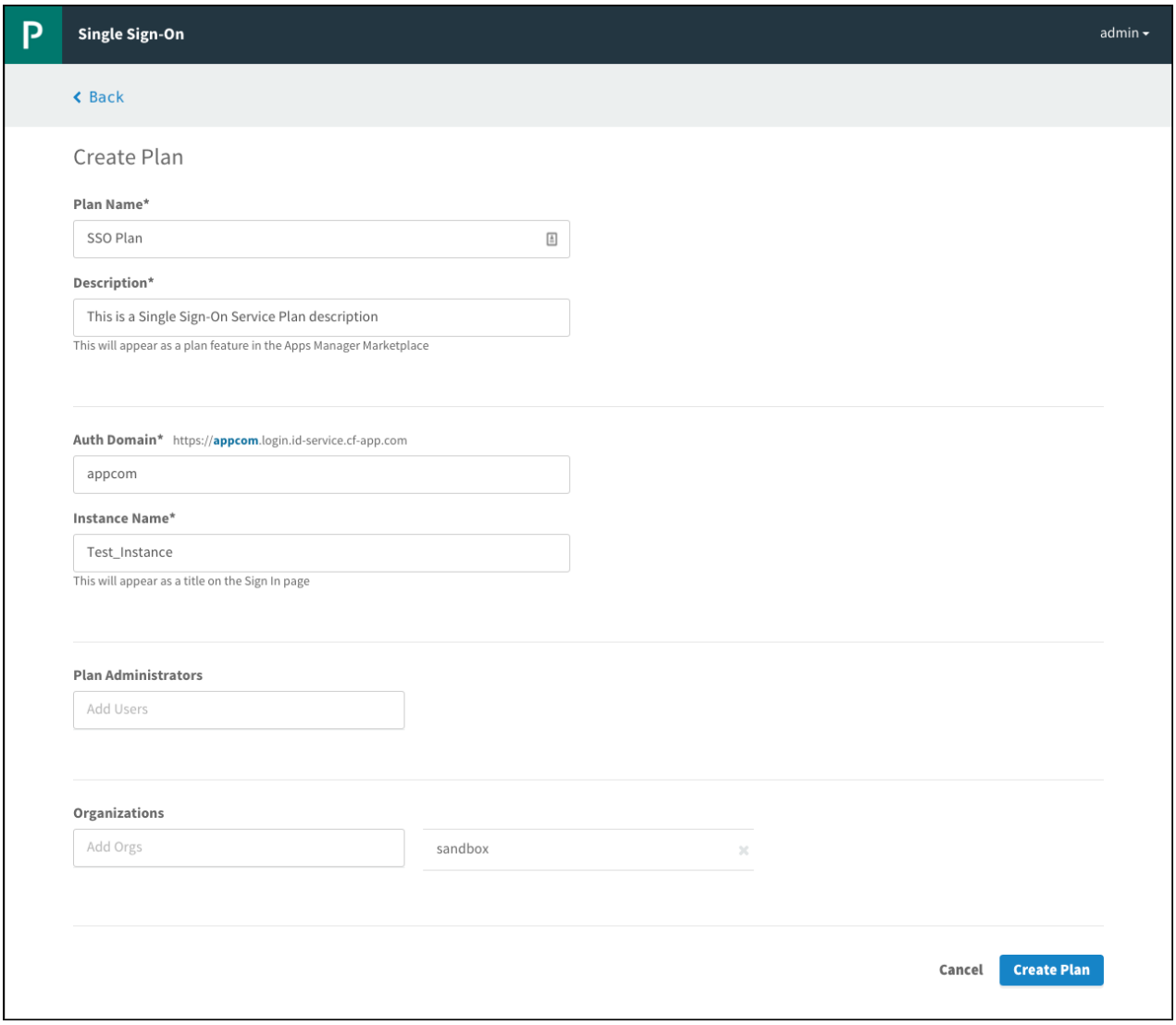
Create or Edit Service Plans

You can use the SSO dashboard to create and configure service plans at any time.

Note: You must create at least one plan for any service before your applications can use it.

1. Log into the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.

2. Click **New Plan** on the SSO dashboard to create a new Single Sign-On service plan.



3. Enter a **Plan Name**.
4. Enter a **Description** to appear as a plan feature in the Services Marketplace.
5. Enter an **Auth Domain** to be the URL where users authenticate to access applications covered by the service plan.
6. Enter an **Instance Name** to appear on the login page and in other user-facing content, such as email communications.
7. Add **Plan Administrators**. These users can view the plan and manage identity providers.
8. Under **Org Visibility**, select which organizations in your Pivotal Cloud Foundry deployment should have access to your Single Sign-On service plan. If you do not select any organizations, the plan will not be available for use and it will not be displayed in the Services Marketplace.
9. Click **Create Plan**. Your new plan appears in the Services Marketplace in the organizations you have selected. Users in those organizations view the plan either in Apps Manager or through the CF CLI by entering `cf marketplace` in a terminal window.

Delete Service Plans

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.
2. Select the name of the plan you want to delete, and click **Edit Plan** in the drop-down menu.
3. Select **Delete** at the bottom of the page.
4. In the popup that appears, click **Delete Plan** to confirm that you want to delete the plan.

Note: This action cannot be undone. Deleting a Single Sign-On service plan removes from the SSO database all of the configurations, identity providers, users, application configurations and resources associated with the plan. It also deletes the associated service instances and service bindings. You must rebind any applications bound to the deleted service instances to new service instances.

Configure a Token Policy

Access tokens carry information about users and clients to servers that manage resources. Servers use access tokens to determine whether the client is authorized or not. Access tokens typically have a short-lived expiration time. *Refresh tokens* carry information necessary to retrieve a new access token after an existing access token expires. Refresh tokens typically have a longer expiration time than access tokens.

Note: The Single Sign-On service allows administrators to override the default expiry of access tokens (12 hours) and refresh tokens (30 days) by zone.

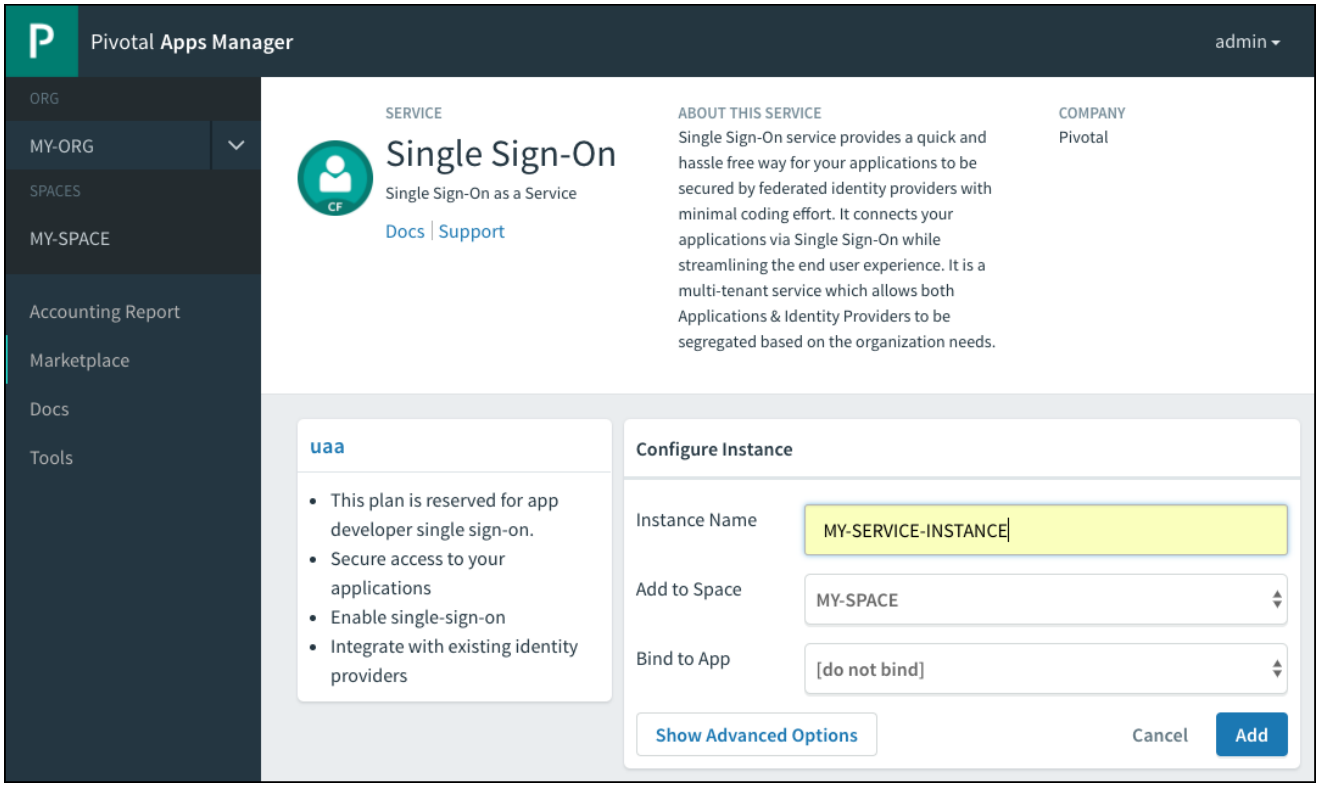
1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.
2. Select the name of the plan you want to configure a token policy for, and click **Configure** in the drop-down menu.
3. Enter the number of seconds for **Access Token Expiration** or select **Use System Default**.
4. Enter the number of seconds for **Refresh Token Expiration** or select **Use System Default**.
5. Click **Save**.

Manage Service Instances

This topic describes how Space Developers create an instance of a Single Sign-On service plan in their space and bind it to an application.

Create Service Instances

1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN` as a Space Developer.
2. Navigate to the organization that the service plan is enabled for.
3. Select **Marketplace** and select the Single Sign-On service you want to create an instance of.
4. Choose your service plan and click **Select this plan**.
5. In the **Configure Instance** box, enter an **Instance Name**.



1. From the **Add to Space** drop-down menu, choose a space for the instance. This space hosts your application. The default is `development`.
2. From the **Bind to App** drop-down menu, choose an application to bind the service instance to. This option defaults to `[do not bind]`. If you do not bind the instance to an app, you can bind it at a later time.
3. Click **Add** to create the service instance.

Delete Service Instances

1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN` as a Space Developer.
2. Navigate to the organization and space that contain the service instance you want to delete.
3. Under **Services** in the space page, find your service instance and click **Delete**.
4. Click **Delete** on the pop-up to confirm that you want to delete the service instance and service bindings.

Note: This action cannot be undone. Deleting a Single Sign-On service instance deletes the configurations on the service instance, as well as the

associated service bindings. You must bind any applications bound to the deleted service instance to a new service instance.

Configure Identity Providers

This topic describes how Pivotal Cloud Foundry (PCF) administrators configure a Single Sign-On (SSO) service plan to manage user access to PCF apps, for users with accounts in the internal user store or with external identity providers.

Configure Internal User Store

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. Find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click **Internal User Store** and select **Edit Provider** from the drop-down menu.
4. (Optional) Under **Authentication Policy** select one of the following:
    - **Disable Internal Authentication**: This option prevents authentication against the internal user store. You must have at least one external identity provider configured.

      Note: The login page does not include the **Email** and **Password** fields if you select this option.

    - **Disable User Management**: This option prevents all users, including administrators, from performing actions on internal users.

      Note: The login page does not include **Create Account** and **Reset Password** links if you select this option.
5. Under **Password Policy Settings**, select **Use Recommended Settings**, **Use Default Settings**, or enter custom settings in the fields below.
6. Click **Save Identity Provider**.

Add Internal Users From the Command Line

You can use the **Internal Users** admin pane to send invitations to users, so that they can add themselves to the internal user store. But you cannot use the admin pane to add users directly.

To create new internal user accounts directly, supplying the user's name, email address and other info, use the UAA Command Line Interface (UAAC) as follows:

1. If you do not already have the UAAC installed, run `gem install cf-uaac` in a terminal window.
2. Create an admin client that can manage users in the Service Plan. Include the following scopes for the client:
    - `clients.admin`
    - `scim.read`
    - `scim.write`
3. Record the **App ID** and **App Secret**. These are used as your client ID and client secret.
4. Target the auth domain of your SSO service plan. This is the URL you provided when creating a Service Plan in the SSO dashboard.

    ```
    $ uaac target https://YOUR-AUTH-DOMAIN.login.YOUR-SYSTEM-DOMAIN
    ```

5. Fetch the **App ID** token for the admin client created above.

    ```
    $ uaac token client get ADMIN-CLIENT-ID
    Client secret:
    ```

6. When prompted with `Client secret`, enter the **App Secret** admin client secret recorded above.
7. Add new users by providing the user's email address, username, and password.

```
$ uaac user add --emails YOUR-USER@EMAIL.COM
User name:  YOUR-USER
Password:  ****
Verify password:  ****
user account successfully added
```

8. (Optional) You can also create groups and add users to them.

```
$ uaac group add
Group name:  YOUR-GROUP
meta
version: 0
created: 2016-02-19T23:17:17.000Z
lastmodified: 2016-02-19T23:17:17.000Z
schemas: urn:scim:schemas:core:1.0
id: 8725b5fd-8da2-4cfc-89b1-c57048f089c2
displayname: YOUR-GROUP
```

To add a member to your new group, use the following command.

```
$ uaac member add YOUR-GROUP YOUR-USER
```

Define Password Policy for the Internal User Store

Administrators can define the password policy for SSO users in the internal user store. The password policy enforces rules that restrict the kinds of passwords users can create.

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.

2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.

3. Click **Internal User Store** and select **Edit Provider** from the drop-down menu.

4. Configure the following under the **Password Complexity** section:

    - **Min Length**: Specify the minimum password length.
    - **Uppercase**: Specify the minimum number of uppercase characters required in a password.
    - **Lowercase**: Specify the minimum number of lowercase characters required in a password.
    - **Special Characters**: Specify the minimum number of special characters required in a password.
    - **Numerals**: Specify the minimum number of numeric characters required in a password.

5. Configure the following under the **Lockout Policy** section:

    - **Failures Allowed**: Specify the number of failed login attempts allowed per hour before a user is locked out.
    - **Lockout Period**: Specify the number of seconds a user is locked out for after excessive failed login attempts.
    - **Password Expires**: Specify the number of months passwords are valid for before users needs to enter a new password.

6. Click **Save Identity Provider**.

Configure Service Provider SAML Settings

For each plan, the Single Sign-On service allows you to configure SAML settings when SAML is used for exchanging authentication and authorization data between the identity provider and the service provider. The SSO service provides the ability to sign authentication requests and require signed assertions from the external identity provider.

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.

2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.

3. Click **Configure SAML Service Provider**.

4. Configure the following settings:

- **Perform signed authentication requests**: The service provider signs requests sent to the external identity provider.
- **Require signed assertions**: The service provider requires that responses from the external identity provider are signed.

5. Click **Save** to save the configurations.

6. Click **Download Metadata**.

Add an External Identity Provider

See the following sets of instructions for how to configure the SSO service to use external identity providers that support SAML 2.0, OpenID Connect (OIDC), and LDAP.

Add a SAML Provider

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.

2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.

3. Click **New Identity Provider**.

4. Enter an **Identity Provider Name**.

5. Select `SAML 2.0` as the Identity Provider Type.

6. Enter a **Description**. This is displayed to Space Developers when they select an identity provider for their app.

7. Enter the external identity provider metadata in one of the following ways:

   - Option 1: Provide the **Identity Provider Metadata URL** and click **Fetch Metadata**.
   - Option 2: Click **Upload Identity Provider Metadata** to upload XML metadata that you downloaded from your external identity provider.

   > **Note**: If you choose to upload the Identity Provider Metadata as an XML file, you will be unable to use the **Fetch Metadata** option to update your Identity Provider metadata later. If metadata changes on the Identity Provider side, you will have to manually re-upload them as an updated XML file.

8. Configure any **User Attributes** to propagate from the identity provider to the service provider. These attributes can include email addresses, first or last names, or external groups. They are sent to apps via OpenID tokens, along with any other stored user information issued by the Single Sign-On service.

   - Select a **User Scheme Attribute** from the drop-down menu.
   - Enter a **SAML Attribute Name** with the corresponding attribute from the incoming SAML assertion.

9. Configure any **Custom Attributes** to propagate from the identity provider to the service provider. These attributes are sent to apps via OpenID tokens issued by the Single Sign-On service.

   - Enter a **Custom Attribute Name**.
   - Enter a **SAML Attribute Name** with the corresponding attribute from the incoming SAML assertion.

10. (Optional) Check **Persist Custom Attributes** if you want to expose custom user attributes through the `/userinfo` endpoint. Your app must also have the `user_attributes` scope assigned in order for the custom attributes to appear.

11. Click **Create Identity Provider** to save the identity provider.

> **Note**: To configure the service provider SAML settings, such as the signing of authentication requests and incoming assertions, click on **Configure SAML Service Provider** on the Identity Providers page.

Add an OIDC Provider

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.

2. Click the plan name and select Manage Identity Providers from the drop-down menu.
3. Click **New Identity Provider**.
4. Enter an **Identity Provider Name**.
5. Enter a **Description**. This is displayed to Space Developers when they select an identity provider for their app.
6. Select `OpenID Connect` as the **Identity Provider Type**.
7. Enter the external OpenID Connect (OIDC) identity provider metadata in one of the following ways:
    - Option 1: Select the **Enable Discovery** checkbox, provide the **Discovery Endpoint URL**, **Relying Party OAuth Client ID**, and **Relying Party OAuth Client Secret** and click **Fetch Scopes**.
    - Option 2: Clear the **Enable Discovery** checkbox and provide the **Authorization Endpoint URL**, **Token Endpoint URL**, **Token Key** (URL), **Relying Party OAuth Client ID**, and **Relying Party OAuth Client Secret**.
8. Select the applicable **Scopes** for the OIDC identity provider.
9. Configure any **User Attributes** to propagate from the identity provider to the service provider. These attributes can include email addresses, first or last names, or external groups. They are sent to apps via OpenID tokens, along with any other stored user information issued by the Single Sign-On service.
    - Select a **User Scheme Attribute** from the drop-down menu.
    - Enter an **ID Token Attribute Name** with the corresponding attribute from the incoming OIDC ID token.
10. Configure any **Custom Attributes** to propagate from the identity provider to the service provider. These attributes are sent to apps via OpenID tokens issued by the Single Sign-On service.
    - Enter a **Custom Attribute Name**.
    - Enter an **ID Token Attribute Name** with the corresponding attribute from the incoming OIDC ID token.
11. (Optional) Check **Persist Custom Attributes** if you want to expose custom user attributes through the `/userinfo` endpoint. Your app must also have the `user_attributes` scope assigned in order for the custom attributes to appear.
12. Click **Create Identity Provider** to save the identity provider.

Add an LDAP Identity Provider

When integrating with an external identity provider for LDAP, authentication becomes chained. An authentication attempt with a user's credentials is first attempted against the internal user store before the external LDAP identity provider. To avoid username collision, do not bootstrap or create users in the UAA directly. You may only have one LDAP external identity provider per service plan.

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click **New Identity Provider**.
4. Enter an **Identity Provider Name**.
5. Enter a **Description**. This is displayed to Space Developers when they select an identity provider for their app.
6. Select `LDAP` as the **Identity Provider Type**. You may only have one LDAP provider per Service Plan.
7. Enter the external LDAP identity provider configurations:
    a. Enter the **Hostname** and **Port**.
    b. Select the applicable **Security protocol**.
    c. Select the applicable **Referral**.
    d. Enter the **User DN** and **Bind Password** for your LDAP service account.
    e. Under the **Users** section, enter the **Search Base**.
    f. Under the **Users** section, you may also enter in **Search Filter** (Optional).
    g. Under the **Users** section, you may select **Just in Time Provisioning**. If this option is enabled, users will be created at login time. If this option is not enabled, users must be created prior to being able to login.
    h. Under the **Groups** section, you may enter in enter the **Search Base** (optional) and **Search Filter** (optional) in order to associate LDAP groups with your user. If you wish to use the `memberOf` attribute on user objects, you can enter in the value `memberOf` as the Search Base instead of

an LDAP path for a group OU, and the Search Filter value will be ignored.

8. Configure any **User Attributes** to propagate from the identity provider to the service provider. These attributes can include email addresses, first or last names, or external groups. They are sent to apps via OpenID tokens, along with any other stored user information issued by the Single Sign-On service.
    - Select a **User Scheme Attribute** from the drop-down menu.
    - Enter an **LDAP Attribute Name** with the corresponding attribute from LDAP.
9. Configure any **Custom Attributes** to propagate from the identity provider to the service provider. These attributes are sent to apps via OpenID tokens issued by the Single Sign-On service.
    - Enter a **Custom Attribute Name**.
    - Enter an **LDAP Attribute Name** with the corresponding attribute from LDAP.
10. (Optional) Check **Persist Custom Attributes** if you want to expose custom user attributes through the `/userinfo` endpoint. Your app must also have the `user_attributes` scope assigned in order for the custom attributes to appear.
11. Click **Create Identity Provider** to save the identity provider.

Delete an External Identity Provider

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click on the name of your external identity provider.
4. Click **Delete** at the bottom of the page.
5. In the popup that appears, click **Delete Identity Provider** to confirm that you want to delete the identity provider, along with all of its configurations.

Note: Deleting an external identity provider deletes all of its configurations. Users will no longer be able to authenticate using the external identity provider. This action cannot be undone.

Configure Group Whitelist for an External Identity Provider

An administrator can include groups from an external identity provider in a Group Whitelist. The list of groups in the whitelist propagates in the ID token when a user authenticates through an external identity provider. An app can then retrieve from the ID token the list of external groups that the user belongs to. An administrator can use these groups to assign permissions by group rather than individual users.

For more details on how to create resource permission mappings, see Create or Edit Resource Permissions.

Note: For an app to retrieve a Group Whitelist containing external groups, the app must request the `roles` scope, and the Group Whitelist must list the external group.

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click on the name of your external identity provider and select **Group Whitelist** from the drop-down menu.
4. Add a group name from your external identity provider.
5. Click **Save Group Whitelist**.

Identity Provider Discovery

This topic describes Identity Provider (IdP) Discovery and how to configure it for your Pivotal Cloud Foundry (PCF) apps that use the Single Sign-On (SSO) service.

What it Does

If users with different email domains access the same PCF app, you can configure SSO to authenticate them through different identity providers.

In this situation, IdP Discovery streamlines the login experience by automatically redirecting the user to their own IdP and shielding them from seeing the IdPs of other app users.

When a user logs in to an app, an account chooser autofills their email address from any previous login, or presents a choice if they have logged in from more than one account. Users can add or remove accounts from the account chooser.

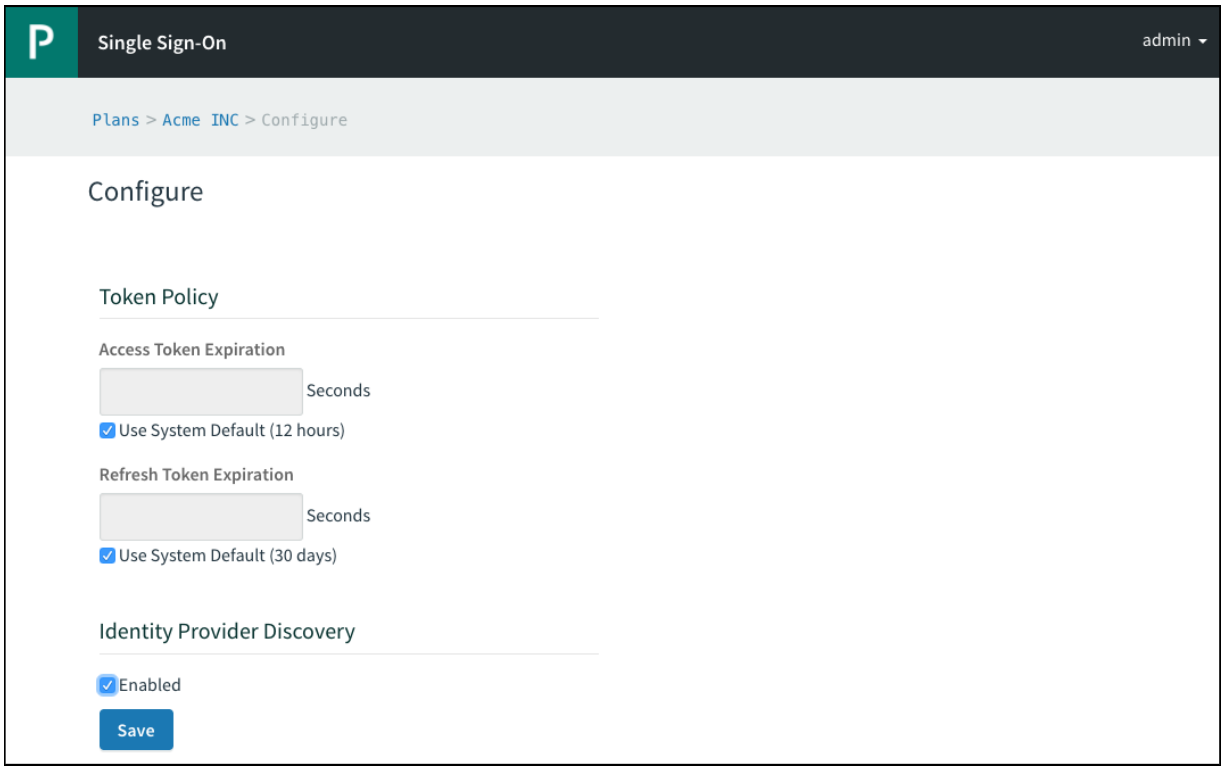
Example

As an example, consider an app used by a company `@company.com` and its competing suppliers `@supplier-1.com` and `@supplier-2.com`. With IdP Discovery, users from all three companies can log in from the same page, and do not have to see or choose from a list of login options that covers all the domains. IdP Discovery ascertains each user's IdP from their email domain.

Enable IdP Discovery

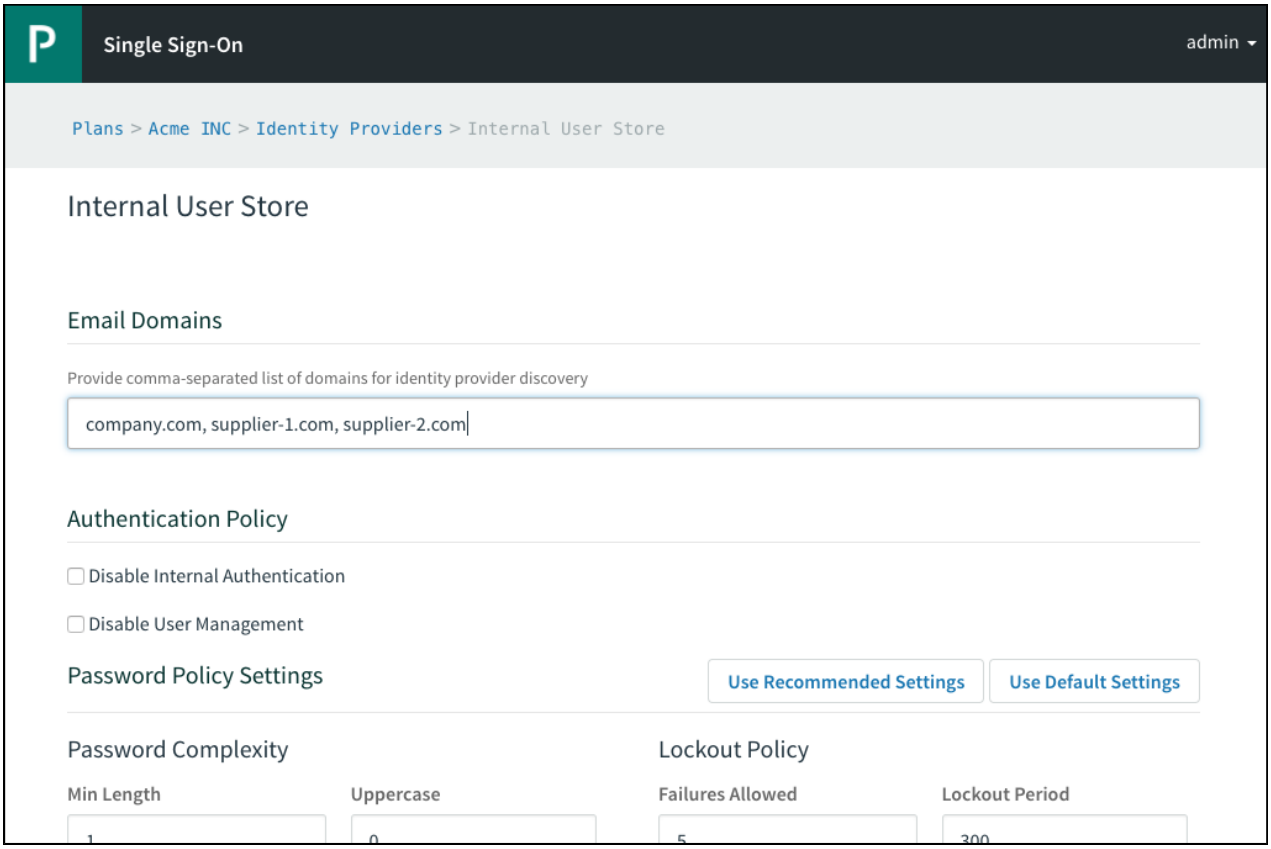
IdP Discovery is associated with a service plan, and configured for the apps bound to instances of that plan. To enable IdP Discovery for a service plan and the apps that use it, you must be a PCF Administrator or a Plan Administrator.

1. Enable IdP Discovery for the SSO Service Plan instance that your app is bound to:
    a. Log into the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the Credentials tab.
    b. Click the plan name and select **Configure** under the plan menu.
    c. Select the checkbox under the **Identity Provider Discovery** section and click **Save**.



2. Click the plan name and select **Manage Identity Providers** under the plan menu.

3. Enter the Email domains you want to include as a comma-separated list under the configuration page for the identity provider plan.



4. In Apps Manager, navigate to your space, open the **Service** tab, and select your service instance.

5. Click the **Manage** link under the service name, and edit the app configuration by selecting the required Identity Providers.

Manage Users

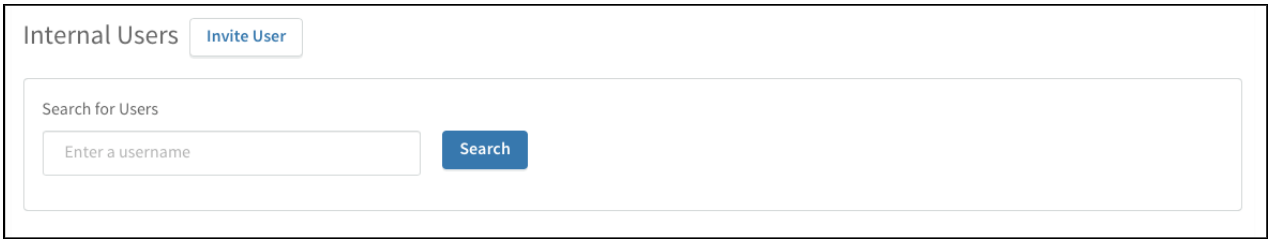
This topic describes how a Pivotal Cloud Foundry (PCF) Plan Administrator uses the Single Sign-On (SSO) service to manage user access to PCF apps, for users with accounts in the internal user store or with external identity providers.

Manage Users in an Internal User Store

The SSO service has an **Internal Users** admin pane that lets you manage user accounts in PCF's internal user store: invite and delete users, request users to reset their passwords, and update user attributes and permissions.

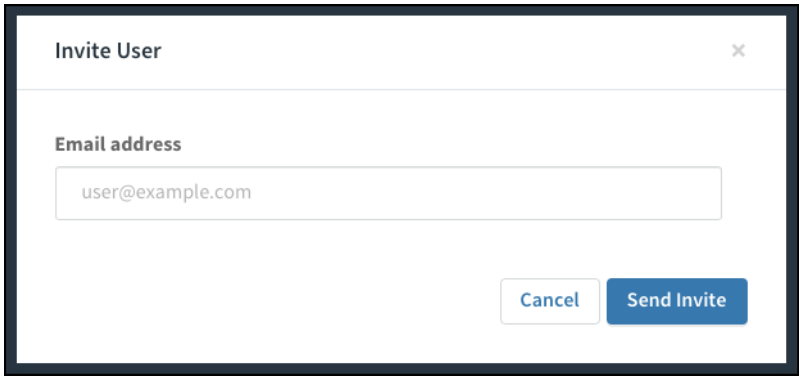
To open the **Internal Users** pane:

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. Find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click **Internal User Store** and select **Internal Users** from the drop-down menu. This brings you to the admin screen.

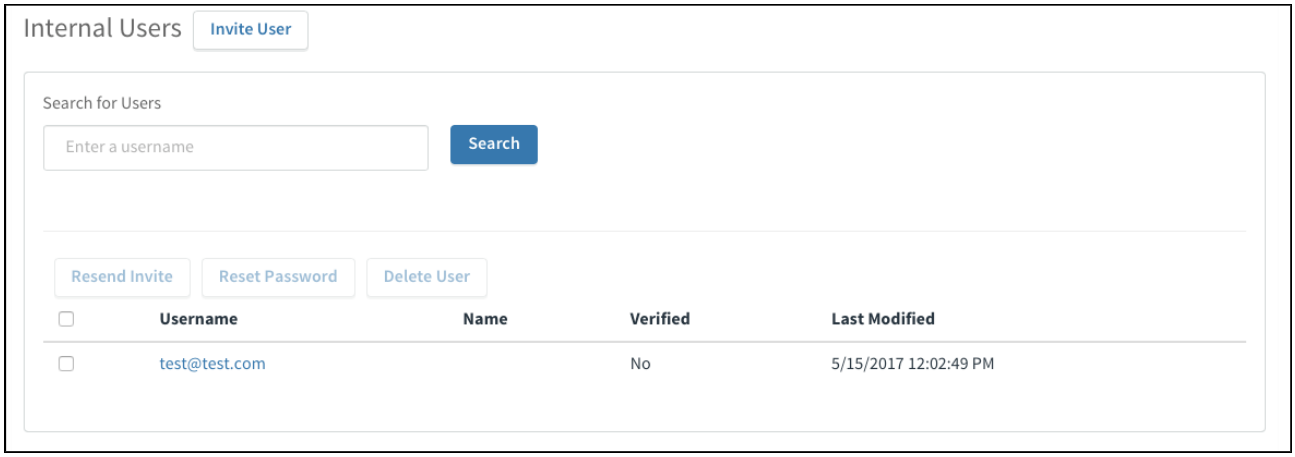


From the **Internal Users** pane, you can:

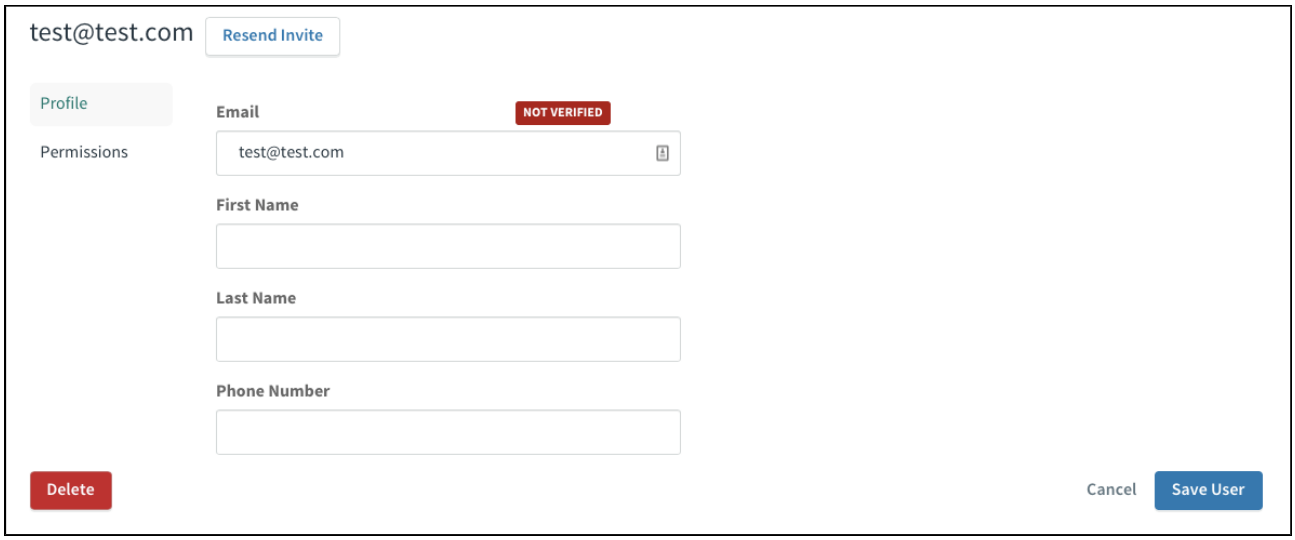
- **Invite users** by clicking **Invite User**, entering their email address, and clicking **Send Invite**.



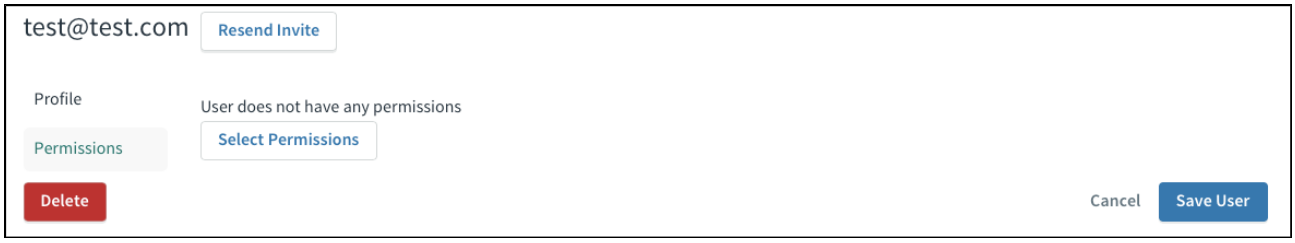
- **Search existing users** by entering a value into the search bar and clicking **Search**. Entering a blank value returns all users in the service plan's internal user store.



- **Resend an invite** to an unverified user by selecting the checkbox next to their username and clicking **Resend Invite**.
- Ask a verified user to **reset their password** by selecting the checkbox next to their username and clicking **Reset Password**.
- **Delete a user** by selecting the checkbox next to their username and clicking **Delete User**.
- **View information about a user** by clicking their username.



- **Update a user profile** including their **Email**, **First Name**, **Last Name**, and **Phone Number** by entering the updated values and clicking **Save User**.
- **View user permissions** by clicking the **Permissions** tab.



- **Update user permissions** by selecting the corresponding permissions and clicking **Save User**.

Manage Users from an External Identity Provider

For each external identity provider that the SSO service connects to, a users admin pane (example: **Okta SSO Users**) lets you browse, delete, and update PCF permissions for user accounts from external identity providers.

To open the external identity provider users admin pane:

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.

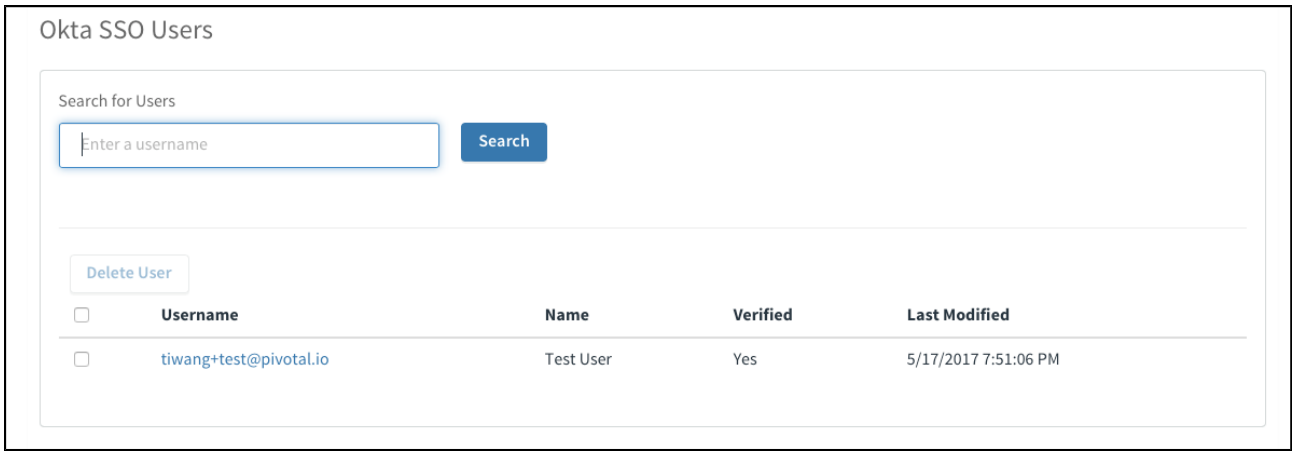
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.

3. Click the external identity provider you want to manage and select the **Users** choice for the provider from the drop-down menu. This brings you to the users admin pane.

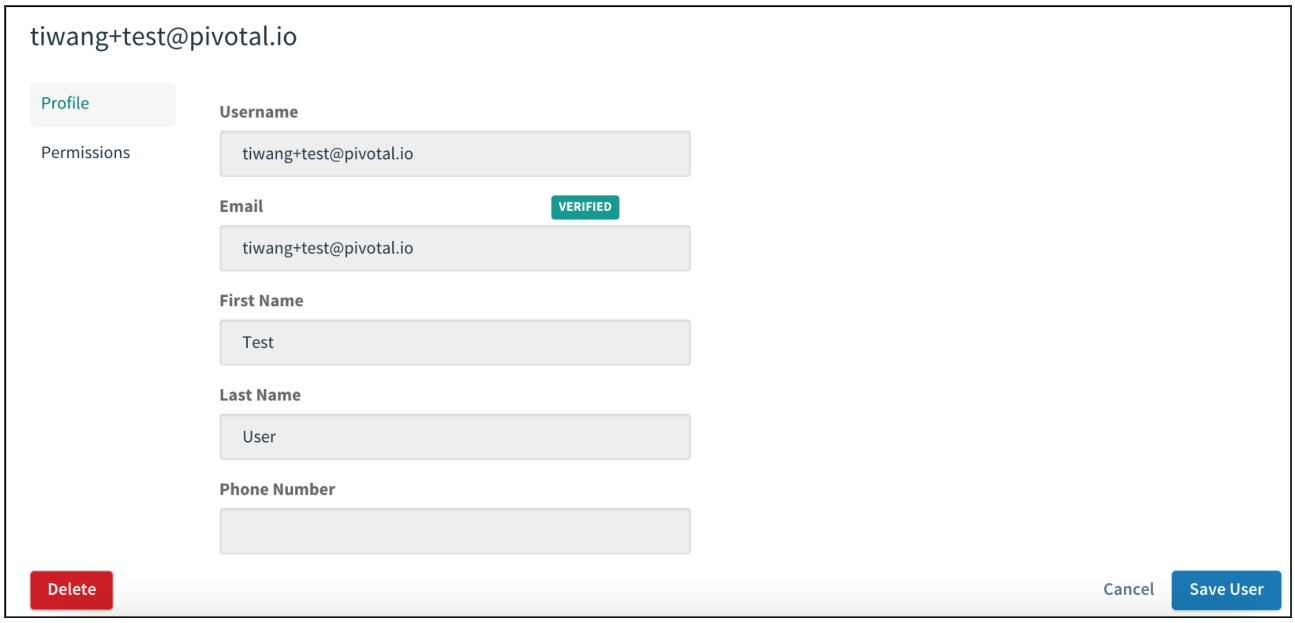
From the external identity provider users admin pane, you can:

- **Search** existing users by entering a value into the search bar and clicking **Search**. Entering a blank value returns all users in the service plan's internal

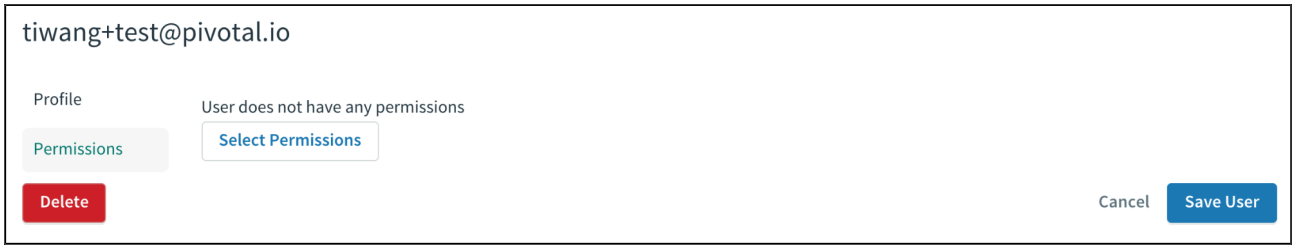
user store.



- **Delete a user** by selecting the checkbox next to their username and clicking **Delete User**.
- **View information about a user** by clicking their username.



- **View user permissions** by clicking the **Permissions** tab.



- **Update user permissions** by selecting the corresponding permissions and clicking **Save User**.

Manage Users with the UAA CLI (UAAC)

You may also use the UAA CLI (UAAC) to manage users for the SSO service. You can use this approach to programatically create new internal users or assign groups (scopes) to any user (whether internal or external). These operations require administrative access through an admin client that must be configured by an administrator for the service plan.

Note: Clients and Groups for SSO should be created directly through the SSO UI or through application manifest bootstrapping. Do not create these through UAAC, as additional metadata is required for their usage by SSO.

1. Install the UAA CLI, `uaac`.

```
$ gem install cf-uaac
```

2. Use the `uaac target AUTH-DOMAIN` command to target your service plan. Auth Domain setting you entered when you created the service plan.

```
$ uaac target my-auth-domain.login.example.com
```

3. Record the **App ID** and **App Secret** from your admin client created using the steps here. You will need to give your admin client `scim.read` to read user information. You can give your admin client either `scim.write` to create users and modify group (scope) memberships or `scim.create` to only create users.

4. Run `uaac token client get ADMIN-APP-ID -s ADMIN-APP-SECRET` to authenticate and obtain an access token for the admin client for your service plan. Replace `ADMIN-APP-ID` with your **App ID** and `ADMIN-APP-SECRET` with your **App Secret**. UAAC stores the token in `~/.uaac.yml`.

```
$ uaac token client get MyAdminAppId -s MyAdminAppSecret
```

5. Use the `uaac contexts` command to display the users and applications authorized by your service plan, and the permissions granted to each user and application. Check that you have the sufficient `scim.write` or `scim.create` permissions under the `scope` section.

```
$ uaac contexts

[1]*[admin]
  client_id: MyAdminAppId
    access_token: aBcdEfg0hIJKlm123.e
    token_type: bearer
    expires_in: 43200
    scope: scim.read scim.write
    jti: 91b3-abcd1233
```

6. Run the following command to create a new internal user: `uaac user add NEW-USERNAME -p NEW-PASSWORD --emails NEW-EMAIL`
Replace `NEW-USERNAME`, `NEW-PASSWORD`, and `NEW-EMAIL` with appropriate information.

```
$ uaac user add Adam -p newSecretPassword --emails adam@example.com
```

7. Run `uaac member add GROUP USERNAME` to add any group to any user (internal or external). Replace `GROUP` and `USERNAME` with appropriate information.

```
$ uaac member add my-app.my-scope Adam
```

8. Run `uaac member delete GROUP USERNAME` to delete any group from to any user (internal or external). Replace `GROUP` and `USERNAME` with appropriate information.

```
$ uaac member delete my-app.my-scope Adam
```

Configure Applications

This topic explains how Pivotal Cloud Foundry (PCF) developers configure their apps to use the Single Sign-On (SSO) service, write SSO integration into their apps, and use the SSO Admin Client to manage connections between SSO identity providers, apps, users and other resources.

Determine Your SSO Application Type

Before you bind or register an app, you must determine its SSO application type and the corresponding OAuth grant type.

If your app authenticates end users, its application type is Web App, Native Mobile App, or Single-Page JavaScript App. If the app does not authenticate end users, but rather accesses other services or APIs on its own behalf, then its type is Service-to-Service App.

See the table below to determine your app's SSO Application Type and OAuth Grant Type:

Application Type	SSO Application Type	OAuth Grant Type
Web	Web App	`authorization code`
Native Mobile, Desktop, or Command Line	Native Mobile App	`password` (the resource owner's password)
Single-Page JavaScript	Single-Page JavaScript App	`implicit`
Service-to-Service	Service-to-Service App	`client_credentials`

Note: The Native Mobile App application type is intended only for highly-trusted apps such as company-owned and managed apps.

Set Up PCF Apps to Use SSO

To configure SSO for an app running internally on PCF, you first need to determine the SSO application type of the app that will use the SSO service.

Then you configure your SSO service for the app using environment variables and bind the app to an SSO service instance. These steps are described below.

Configure SSO Properties

The SSO service reads its configuration properties from environment variables that are set in the apps that use it. To enable a PCF-hosted app to bind to an SSO service instance, you must set the following environment variables:

- `GRANT_TYPE` — the type of OAuth authentication associated with the SSO Application Type in the table above.
- `SSO_IDENTITY_PROVIDERS` — the internal or external identity provider(s) for the app to use.

You can set additional environment variables to further configure how an app uses SSO, as described below. Most of these environment variables are prefixed with `SSO_`.

There are two ways to set the SSO configuration properties for an app:

- Set the environment variables manually after you deploy the app, in Apps Manager or with the Cloud Foundry Command-Line Interface (cf CLI).
- Include the config settings in the application manifest, so that PCF bootstraps them automatically when it deploys the app.

Note: These configurations are only applied when binding to the service instance. A `cf push` of the app does not update the configurations. To update these configurations, manually update them using the SSO dashboard or unbind and rebind the service instance. In the SSO sample applications, the manifest binds the app to a service instance on the initial push using the manifest value `services: - sample-instance`.

Manually Configure Apps for SSO

For apps already deployed to PCF, you can set their `GRANT_TYPE`, `SSO_IDENTITY_PROVIDERS`, and other SSO configuration environment variables with the `cf set-env` command, or in Apps Manager as follows:

1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN`.
2. Navigate to your app.
3. Click the **Env Variables** tab.
4. Click **Add an Env Variable**.
5. For **Variable Name** enter the name of the SSO configuration property that you are setting, such as `GRANT_TYPE`.
6. For **Value**, enter the property value. For example, to set the `GRANT_TYPE` property for a Single-Page JavaScript App, enter `implicit`, which is the OAuth Grant Type listed for your SSO application type above.
7. Bind and restage your app.

Bootstrap SSO Configuration

In SSO v1.4.0 and later, you can include SSO configuration properties in your application manifest, to automatically bootstrap the values when you bind or rebind the app to an SSO service instance.

The values from the manifest automatically save to the environment variables that configure your app for SSO. Bootstrapping SSO configuration values from the manifest eliminates the need to set environment variables after you deploy your app.

Note: These configurations are only applied when binding to the service instance. A `cf push` of the app does not update the configurations. To update these configurations, manually update them using the SSO dashboard or unbind and rebind the service instance. In the SSO sample applications, the manifest binds the app to a service instance on the initial push using the manifest value `services: - sample-instance`.

This snippet below shows how to include `GRANT_TYPE` `SSO_IDENTITY_PROVIDERS` in your manifest.

```
---
applications:
  - name: APPLICATION NAME
    env:
      GRANT_TYPE: password
      SSO_IDENTITY_PROVIDERS: uaa, sample-identity-provider
```

The `GRANT_TYPE` property defaults to `authorization_code`, for Web App application type. `SSO_IDENTITY_PROVIDERS` defaults to `uaa`, for the PCF internal user store.

If you specify your own scopes and authorities, consider including the following values in your `SSO_SCOPES` or `SSO_AUTHORITIES` property list. These values are not added your user-provided list by default:

- `openid` — for apps with `authorization code`, `password`, and `implicit` grant type
- `uaa.resource` — for apps with `client_credentials` grant type

The table below lists all SSO properties that you can set in your application manifest to bootstrap the values into your app's SSO client configuration.

After an app deploys with bootstrapped SSO configuration values, it is ready to bind to an SSO service instance.

SSO Configuration Properties

The table below provides descriptions and default values for environment variables that apps use to configure SSO. See the SSO sample applications for details, and the `manifest.yml` files in the same repo for examples of bootstrapping these values.

Property Name	Description	Default
name	Name of the app	(N/A - Required Value)
`GRANT_TYPE`	Allowed grant type for the app through the SSO service. Only one grant type per app is supported by SSO.	`authorization_code`

Property Name	Description	Default
`SSO_IDENTITY_PROVIDERS`	Allowed identity providers for the app through the SSO service plan. This is a comma-separated list of identity provider origin keys. The origin keys are derived from the identity provider name using the following rules: • Uppercase letters are converted to lowercase letters. • Spaces are converted to hyphens. • Periods are converted to hyphens. For example, if your identity provider name is `example.com Provider`, the corresponding origin key is `example-com-provider`.	`uaa`
`SSO_REDIRECT_URIS`	Comma-separated whitelist of redirection URIs allowed for the app. Each value must start with `http://` or `https://`.	(Always includes the app route)
`SSO_SCOPES`	Comma-separated list of scopes that belong to the app and are registered as client scopes with the SSO service. This value is ignored for client credential grant type apps.	`openid`
`SSO_AUTO_APPROVED_SCOPES`	Comma-separated list of scopes that the app is automatically authorized when acting on behalf of users through SSO service.	(Defaults to existing scopes/authorities)
`SSO_AUTHORITIES`	Comma-separated list of authorities that belong to the app and are registered as client authorities with the SSO service. Privileged identity zone/plan administrator scopes, such as `scim.read`, `idps.write` cannot be bootstrapped and must be assigned by zone/plan administrators. This value is ignored for any grant type other than client credentials.	`uaa.resource`
`SSO_REQUIRED_USER_GROUPS`	Comma-separated list of groups a user must have in order to authenticate successfully for the app.	(No value)
`SSO_ACCESS_TOKEN_LIFETIME`	Lifetime in seconds for the access token issued to the app by the SSO service.	43200
`SSO_REFRESH_TOKEN_LIFETIME`	Lifetime in seconds for the refresh token issued to the app by the SSO service.	2592000 (not used for client credentials)
`SSO_RESOURCES`	Resources that the app will use as scopes/authorities for the SSO service to be created during bootstrapping if they do not already exist. The input format can be referenced in the provided sample manifest. Note that currently all permissions within the same top level permission, such as `todo.read` and `todo.write`, must be specified in the same application manifest. Currently you cannot specify additional permissions in the same top level permission, such as `todo.admin`, in additional application manifests.	(No value)
`SSO_ICON`	App icon that will be displayed next to the app name on the Pivotal Account dashboard if show on home page is enabled. Do not exceed 64kb.	(No value)
`SSO_LAUNCH_URL`	App launch URL that will be used for the app on the Pivotal Account dashboard if show on home page is enabled.	(Application route)
`SSO_SHOW_ON_HOME_PAGE`	If set to true, the app will appear on the Pivotal Account dashboard with the corresponding icon and launch URL.	True

Additional information and manifest examples are available on the identity sample apps.

Remove SSO Configuration Properties

You can remove SSO configuration properties for an app, or any environment variables set through `cf set-env`, Apps Manager, or bootstrapping as follows:

1. Run `cf unset-env APP_NAME PROPERTY_NAME`.
2. Rebind the app.

Bind a PCF App

After a PCF app is configured for SSO, you can bind it to an SSO service instance as follows:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your app runs.
3. Under **Applications**, click the name of your app.
4. Click the **Services** tab.
5. Click **Bind a Service**.
6. Bind your app to a service to create an associated OAuth Client.
    a. Select an existing SSO service instance from the drop-down menu and click **Bind**.
    b. Create a new service instance:
        i. Click **or add from Marketplace**.
        ii. Select the **Single Sign-On** service under Services Marketplace.
        iii. Select a Service Plan, then click **Select this plan**.
        iv. Enter an **Instance Name**, select a space, select an app, then click **Add**.
7. Click **Manage** under the SSO service instance to launch the SSO dashboard.
8. Click your app.
9. Specify a value in the **App Launch URL** field that you want to set as the address of your app.
10. Upload an app icon for your app.
11. Click **Show on homepage** to display the app on the UAA or Pivotal Account home page.

> **Note**: If you would like app to display on the home page, you must enter an **App Launch URL** or upload an app icon.

12. Select one or more **Identity Providers** for your app. Internal User Store is the default.

> **Note**: When binding a PCF app, a Space Developer can choose from internal and external identity providers. If the Space Developer selects multiple identity providers, users must select which provider to use when they sign in. This option is available for all application types except `Service-to-Service App`.

13. If your Application Type is `Web App` or `Single-Page JavaScript App`, enter a whitelist of **Auth Redirect URIs** beneath **Redirect URIs**. The redirect query parameter specified on the OAuth request must match the URIs specified in this list. Otherwise, SSO rejects the request.
14. For the **Scopes** field, specify the permissions that the app can request on the user's behalf. This field defaults to `openid` for Web, Native Mobile, and Single-Page JavaScript Apps. This field defaults to `uaa.resource` for Service-to-Service Apps. If this app is purely for authentication purposes, then the `openid` scope is sufficient. If the app makes API calls on behalf of the end user, specify both the scopes enforced by the API and the scopes to be requested by the app.

Scope	Description
`openid`	Provides access to make OpenID Connect requests
`user_attributes`	Provides access to custom attributes from an external identity provider
`roles`	Provides access to external groups from an identity provider
`uaa.resource`	Provides access to the check_token endpoint for service-to-service flows

> **Note**: Under **Scopes**, you can select resources defined in any space if the application type is a `Web App`, `Native Mobile App`, or `Single-Page JavaScript App`. If the application type is a `Service-to-Service App`, you can only select resources defined within the space.

1. For **Auto-Approved Scopes**, select any scopes that the SSO service automatically approves when the app makes a request on behalf of a user. Select only scopes pertaining to apps owned and managed by your company. Do not select scopes that pertain to apps external to PCF.
2. Click **Save Config**. The **Next Steps** page appears, describing the endpoints required for app integration. For more information, see Integrate SSO with Apps below.

Manage App Configurations via SSO Dashboard

The SSO dashboard allows application developers to view the app configurations and resources available within their space. To access the dashboard, first you must create a service instance for your Space. Then you can follow the steps below to manage your application configurations via the SSO dashboard.

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click **Manage** next to the SSO service instance. This launches the SSO dashboard.

Register an External App

1. Determine the type of the app that will use the SSO service.
2. Log in to Apps Manager as a Space Developer.
3. Select the space where your service instance is located.
4. Under **Services**, click **Manage** next to the SSO service instance. This launches the SSO dashboard.
5. Click **New App**.
6. Enter an **App Name**.
7. Choose an app type under **Select an Application Type**.
8. Enter an **App Launch URL** that specifies the address of your app.
9. Upload an app icon for your app.
10. Click **Show on homepage** to display the app on the UAA or Pivotal Account home page.

> **Note**: To display the app on the home page, you must enter an **App Launch URL** or Upload an app icon.

11. Select one or more **Identity Providers** for your app. Internal User Store is the default.

> **Note**: When registering an externally-hosted app, a Space Developer can choose from internal and external identity providers. If the Space Developer selects multiple identity providers, users must select which provider to use when they sign in. This option is available for all application types except `Service-to-Service App`.

12. If your Application Type is `Web App` or `Single-Page JavaScript App`, enter a whitelist of **Auth Redirect URIs** beneath **Redirect URIs**. The redirect query parameter specified on the OAuth request must match the URIs specified in this list. Otherwise, SSO rejects the request.
13. For the **Scopes** field, specify the permissions that the app can request on the user's behalf. This field defaults to `openid` for Web, Native Mobile, and Single-Page JavaScript Apps. This field defaults to `uaa.resource` for Service-to-Service Apps. If this app is purely for authentication purposes, then the `openid` scope is sufficient. If the app makes API calls on behalf of the end user, you must specify both the scopes enforced by the API and the scopes to be requested by the app.

Scope	Description
`openid`	Provides access to make OpenID Connect requests
`user_attributes`	Provides access to custom attributes from an external identity provider
`roles`	Provides access to external groups from an identity provider
`uaa.resource`	Provides access to check_token endpoint for service-to-service flows

> **Note**: Add the `user_attributes` scope to the client scopes to return user attributes from the ID token.

> **Note**: Under **Scopes**, you can select resources defined in any space if the application type is a `Web App`, `Native Mobile App`, or `Single-Page JavaScript App`. If the application type is a `Service-to-Service App`, you can only select resources defined within the space.

14. For **Auto-Approved Scopes**, select any scopes that the SSO service automatically approves when the app makes a request on behalf of a user. Select only scopes pertaining to apps owned and managed by your company. Do not select scopes that pertain to apps external to PCF.

15. Click **Create App**. The **Next Steps** page appears, describing the endpoints required for app integration. For more information, see Integrate SSO with Applications below.

Integrate SSO with an App

Because SSO service is based on the OAuth protocol, any app that uses SSO must be OAuth-aware.

Java Apps

If you are using Java, see Single Sign-On Service Sample Applications. These are sample apps created using Spring Boot for all four application types. These apps use the SSO Service Connector, which auto-configures the app for OAuth. After binding the app to an SSO service instance, you must restart the app for the new SSO configuration to take effect.

Non-Java Apps

To configure non-Java apps for OAuth, supply the following properties as environment variables to your app after the SSO service bind. You can view this information on the **Next Steps** page of the SSO dashboard.

- **App ID**, also known as OAuth Client ID
- **App Secret**, also known as OAuth Client Secret
- **OAuth Authorization URL**, the endpoint for client authorization
- **OAuth Token URL**, the endpoint for token retrieval

To validate the token, you must verify the following:

1. The token is a properly signed JSON Web Token with an appropriate public key. The key can be downloaded from the **Token Verification Key** endpoint specified on the **Next Steps** page.

2. The value of `aud` in the token matches your **App ID**.

3. The value of `iss` matches `https://AUTH-DOMAIN.uaa.YOUR-SYSTEM-DOMAIN/oauth/token`.

4. The expiry time of the token, `exp`, has not passed.

Create Admin Client

You can create an admin client to perform administrative functions, such as manage identity providers, apps, users, groups, and resources in a specific zone where you create the client.

You must be at least a plan administrator to perform these steps.

Create an admin client as follows:

1. Log in to Apps Manager.

2. Select the space where your service instance is located. This specifies the zone you manage as an admin client.

3. Under **Services**, click the **Single Sign-On** service.

4. Click **Manage** next to your SSO service instance to launch the SSO dashboard.

5. Click **New App**.

6. Enter an **App Name**.

7. Under **Select an Application Type**, select **Service-to-Service App**.

8. Click **Select Scopes** and choose what actions the admin client can perform from the following **Admin Permissions**:

Scope	Description
`clients.admin`	Provides superuser access to create, modify, and delete clients
`clients.read`	Provides access to read information about clients
`clients.write`	Provides access to create and modify clients
`scim.create`	Provides access to create users
`scim.read`	Provides access to read information about users and group memberships
`scim.write`	Provides access to create, modify, and delete users and group memberships
`idps.read`	Provides access to read information about identity providers
`idps.write`	Provides access to create, modify, and delete identity providers

9. Click **Create App**.

Delete App that Uses SSO

Delete a PCF app or an external app that uses SSO as follows:

Delete a PCF App

To delete an app hosted on PCF:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your app is located.
3. Under **Applications**, click the name of your app.
4. On the Application page, click **Delete App**.
5. On the popup, click **Delete** to confirm that you want to delete the app and its configurations from Apps Manager and the service dashboard.

Delete an External App

To delete an external app that uses SSO:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click **Manage** next to your SSO service instance to launch the SSO dashboard.
4. Click your app.
5. Click **Delete** at the bottom of the page.
6. On the popup, click **Delete App** to confirm that you want to delete the app and its configurations.

> **Note**: Deleting an externally hosted app in PCF removes the app and its configurations from the SSO dashboard. However, it still exists on your hosted platform.

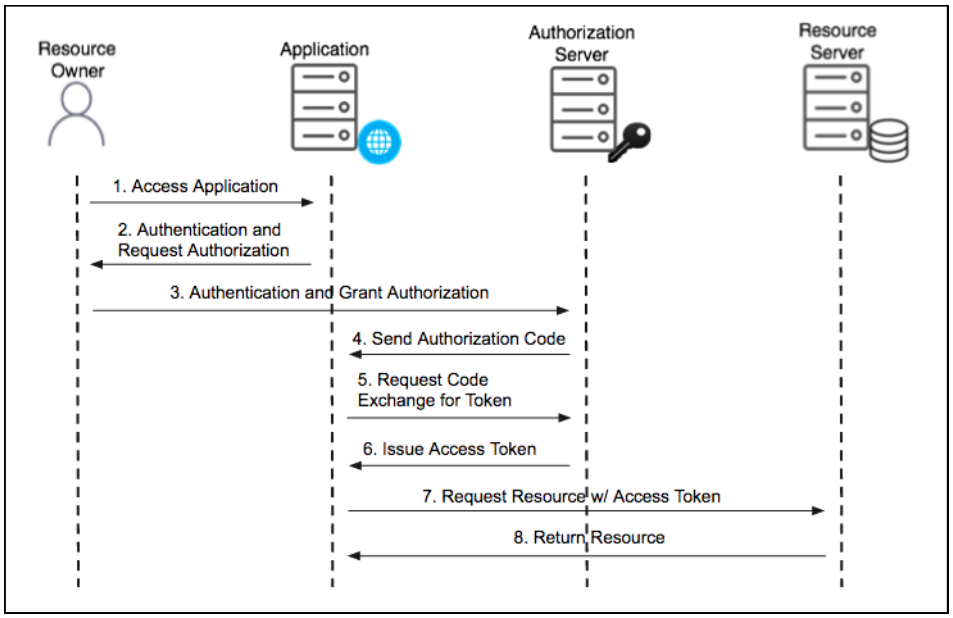
Web App

This topic describes the OAuth 2.0 Authorization Code grant type supported by Pivotal Single Sign-On (SSO). The authorization code grant type is the most commonly used grant type. This grant type is for server-side applications.

OAuth 2.0 Roles

- **Resource Owner**: A person or system capable of granting access to a protected resource.
- **Application**: A client that makes protected requests using the authorization of the resource owner.
- **Authorization Server**: The Single Sign-On server that issues access tokens to client applications after successfully authenticating the resource owner.
- **Resource Server**: The server that hosts protected resources and accepts and responds to protected resource requests using access tokens. Applications access the server through APIs.

Authorization Code Flow



1. **Access Application**: The user accesses the application and triggers authentication and authorization.
2. **Authentication and Request Authorization**: The application prompts the user for their username and password. The first time the user goes through this flow for the application, the user sees an approval page. On this page, the user can choose permissions to authorize the application to access resources on their behalf.
3. **Authentication and Grant Authorization**: The authorization server receives the authentication and authorization grant.
4. **Send Authentication Code**: After the user authorizes the application, the authorization server sends an authorization code to the application.
5. **Request Code Exchange for Token**: The application receives the authorization code and requests an access token from the authorization server. This gives the application access to the approved permissions.
6. **Issue Access Token**: The authorization server validates the authorization code and issues an access token.
7. **Request Resource w/ Access Token**: The application attempts to access the resource from the resource server by presenting the access token.
8. **Return Resource**: If the access token is valid, the resource server returns the resources that the user authorized the application to receive.

The resource server runs in PCF under a given space and organization. Developers set the permissions for the resource server API endpoints. To do this, they create resources that correspond to API endpoints secured by the Single Sign-On service. Applications can then access these resources on behalf of users.

Native Mobile App

For Native Mobile and Desktop applications, Pivotal Single Sign-On (SSO) supports the Resource Owner Password OAuth 2.0 grant type. This password grant type is for highly trusted applications where resource owners share their credentials directly with the application.

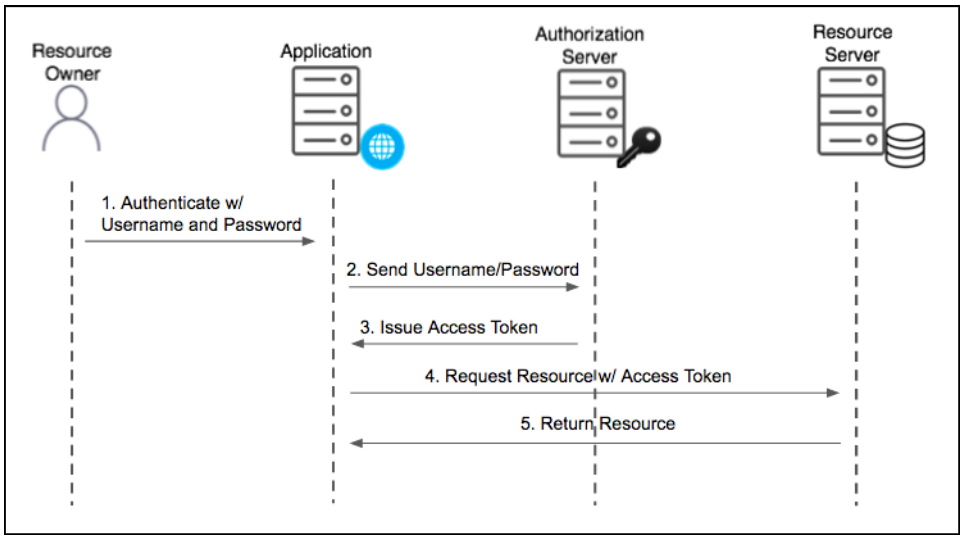
OAuth 2.0 Roles

The following roles are available in an OAuth 2.0 scenario:

- **Resource Owner**: A person or system capable of granting access to a protected resource.
- **Application**: A client that makes protected requests using the authorization of the resource owner.
- **Authorization Server**: The Single Sign-On server that issues access tokens to client applications after successfully authenticating the resource owner.
- **Resource Server**: The server that hosts protected resources and accepts and responds to protected resource requests using access tokens. Applications access the server through APIs.

Native Mobile App Flow

The following diagram shows the authentication flow used by mobile apps. In this scenario, the application is backed by a resource server and both are secured by the UAA authorization server.



1. **Authenticate w/ Username and Password**: The user authenticates with the application using their username and password.
2. **Send Username/Password**: The application sends the username and password to the authorization server for validation.
3. **Issue Access Token**: The authorization server validates the username and password and issues an access token.
4. **Request Resource w/ Access Token**: The application attempts to access the resource from the resource server by presenting the access token.
5. **Return Resource**: If the access token is valid, the resource server returns the resources that the user authorized the application to receive.

The resource server runs in PCF under a given space and organization. Developers set the permissions for the resource server API endpoints. To do this, they create resources that correspond to API endpoints secured by the Single Sign-On service. Applications can then access these resources on behalf of users.

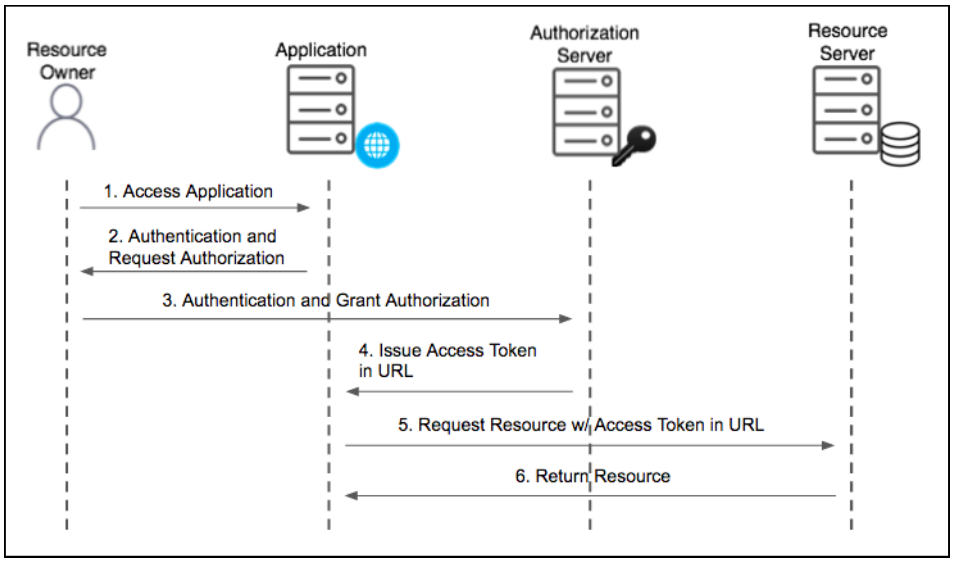
Single-Page Javascript App

This topic describes the OAuth 2.0 implicit grant type supported by Pivotal Single Sign-On (SSO). The implicit grant type is for applications with a client secret that is not guaranteed to be confidential.

OAuth 2.0 Roles

- **Resource Owner**: A person or system capable of granting access to a protected resource.
- **Application**: A client that makes protected requests using the authorization of the resource owner.
- **Authorization Server**: The Single Sign-On server that issues access tokens to client applications after successfully authenticating the resource owner.
- **Resource Server**: The server that hosts protected resources and accepts and responds to protected resource requests using access tokens. Applications access the server through APIs.

Implicit Flow



1. **Access Application**: The user accesses the application and triggers authentication and authorization.
2. **Authentication and Request Authorization**: The application prompts the user for their username and password. The first time the user goes through this flow for the application, the user sees an approval page. On this page, the user can choose permissions to authorize the application to access resources on their behalf.
3. **Authentication and Grant Authorization**: The authorization server receives the authentication and authorization grant.
4. **Issue Access Token**: The authorization server validates the authorization code and returns an access token with the redirect URL.
5. **Request Resource w/ Access Token in**: The application attempts to access the resource from the resource server by presenting the access token in the URL.
6. **Return Resource**: If the access token is valid, the resource server returns the resources that the user authorized the application to receive.

The resource server runs in PCF under a given space and organization. Developers set the permissions for the resource server API endpoints. To do this, they create resources that correspond to API endpoints secured by the Single Sign-On service. Applications can then access these resources on behalf of users.

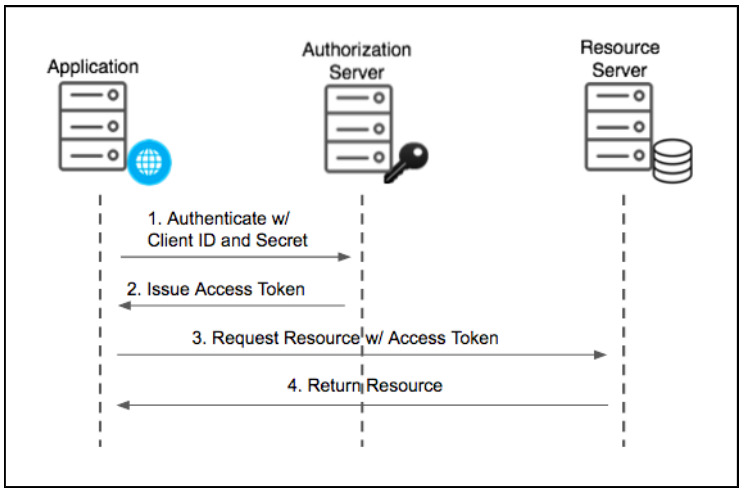
Service-to-Service App

For Service-to-Service applications, Pivotal Single Sign-On (SSO) supports the Client Credentials OAuth 2.0 grant type. The client credentials grant type is for applications that can request an access token and access resources on its own. This is often the case when there are services that call APIs without users.

OAuth 2.0 Actors

- **Application**: A client that makes protected requests using the authorization of the resource owner.
- **Authorization Server**: The Single Sign-On server that issues access tokens to client applications after successfully authenticating the resource owner.
- **Resource Server**: The server that hosts protected resources and accepts and responds to protected resource requests using access tokens. Applications access the server through APIs.

Client Credentials Flow



1. **Authenticate w/ Client ID and Secret**: The application authenticates with the authorization server using its client ID and client secret.
2. **Issue Access Token**: The authorization server validates the client ID and client secret and issues an access token.
3. **Request Resource w/ Access Token**: The application attempts to access the resource from the resource server by presenting the access token.
4. **Return Resource**: If the access token is valid, the resource server returns the resources to the application.

The resource server runs in PCF under a given space and organization. Developers set the permissions for the resource server API endpoints. To do this, they create resources that correspond to API endpoints secured by the Single Sign-On service. Administrators can create admin clients to perform automated management actions without a user. See Create Admin Client.

Manage Resources

This topic describes how a Space Developer defines resources required by an app bound to a Single Sign-On (SSO) service instance and how an administrator grants resource permissions.

In this topic, *resources* are the API endpoints that users and apps need to retrieve information from a resource server. After an administrator creates resources, they assign the resources to users and apps. Users can then grant apps access to the resources, for example to query API endpoints on their behalf.

Because developers know what endpoints exist for their apps, they are responsible for creating resources.

Create or Edit Resources

If an app requires access to specific resources such as API endpoints, permissions for those resources must be either bootstrapped from the application manifest or defined by the Space Developer in the SSO dashboard.

To bootstrap resources from the manifest, follow the instructions in the SSO Sample Applications repo.

To create resources in the SSO Dashboard:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click **Manage** next to your SSO service instance to launch the SSO dashboard.
4. Click the **Resources** tab.
5. Click **New Resource**.
6. Enter a **Resource Name**.
7. Create **Permissions** that the OAuth client for your app needs to access from the resource server.
    a. Enter one or more **Attributes** or **Actions** for each permission.
    b. Enter a **Description** for each permission.
8. Click **Save Resource**. The administrator must create resource permissions so that users can access the resource. For more information, see Create or Edit Resource Permissions below.

Note: Space Developers create resources within a space. Space Developers only see the resources created in the spaces they have access to and can only assign those to the apps in those spaces.

Delete Resources

1. Log in to Apps Manager as a Space Developer.
2. Click the **Manage** link under the SSO service instance to launch the service dashboard.
3. Click the **Resources** tab.
4. Click the resource to delete.
5. Click **Delete** at the bottom of the page.
6. On the popup, click **Delete Resource** to delete the resource.

Note: Deleting a resource removes it from the permission mappings and from the app. You must reconfigure the updated permissions in both areas.

Create or Edit Resource Permissions

After a Space Developer defines resources required by an app, an administrator must grant access to those resources. SSO allows administrators to map groups of users from the identity provider to the resource permissions defined by the Space Developer.

Once resource permissions mappings are configured, when a user authenticates and obtains a token, the user's group memberships will automatically be mapped into scopes that are directly included in the token.

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click the name of the external identity provider you want to define permissions for and select **Resource Permissions** from the drop-down menu.
4. Click **New Permissions Mapping**.
5. Enter a **Group Name**.
6. Click **Select Permissions** to choose the permissions that users in the group should have access to.
7. Click **Save Permissions Mapping**.

Note: Groups with unsupported characters in Permission Mappings are not editable.

Delete Resource Permissions

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your User Account and Authentication (UAA) administrator credentials. You can find these credentials in your Pivotal Elastic Runtime tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.
3. Click on the name of the external identity provider you want to define permissions for and select **Resource Permissions** from the drop-down menu.
4. Click the group name of the resource permission you want to delete.
5. Click **Delete** at the bottom of the page.
6. On the popup, click **Delete Permissions Mapping** to delete the resource.

Note: Groups with unsupported characters in Permission Mappings are not editable.

About Space Protection for Resources

OAuth 2.0 provides the concept of a *scope* in order to limit the amount of access that is granted to an access token. A scope is the intersection of a user's groups and a client's scopes.

For a user to gain access to a resource, they must meet the following conditions, which can only be set up by plan administrators:

- The user must be assigned the resource as a group. For information on how to do this, see Manage Users.
- The user must access an app that has the resource assigned as a scope.

App developers can assign scopes to any app that is *not* a service-to-service app. But, only plan administrators can assign scopes to users.

When assigning a resource as a scope for a service-to-service app, app developers can only assign resources they have created within their own space. Only an plan administrator can assign a scope from another space to a service-to-service app.

Active Directory Federation Services Integration Guide Overview

Active Directory Federation Services (AD FS) is a standards-based service that securely shares identity information between applications. This documentation describes how to configure a single sign-on partnership between AD FS as the Identity Provider (IdP) and the Single Sign-On Service (SSO) for Pivotal Cloud Foundry as the Service Provider (SP).

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

Prerequisites

To integrate AD FS with Pivotal Cloud Foundry (PCF), you need the following:

Pivotal

- PCF, version 1.7.0 or later
- Single Sign-On, version 1.1.0 or later

Active Directory Federation Services

- Active Directory Federation Services subscription
- A user with Administrative privileges

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic.

Active Directory Federation Services Integration Guide

Configuring AD FS with SSO

Complete both steps below to integrate your deployment with AD FS and SSO.

1. Configure Active Directory Federation Services as an Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

- Testing
- Troubleshooting

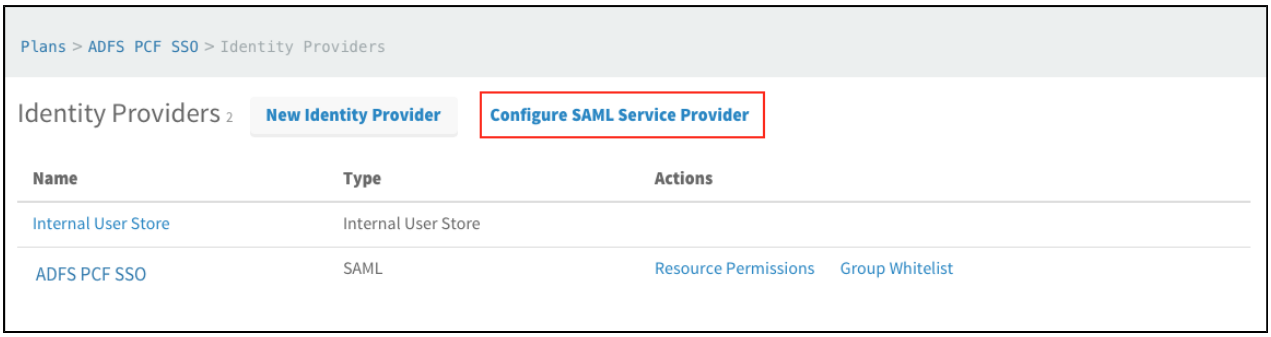
Configure Active Directory Federation Services as an Identity Provider

This topic describes how to set up Active Directory Federation Services (ADFS) as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and ADFS.

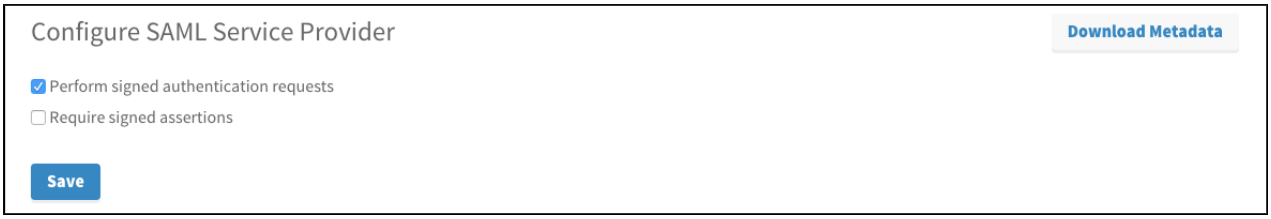
Set Up SAML in PCF

1. Log in to the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **Configure SAML Service Provider**.



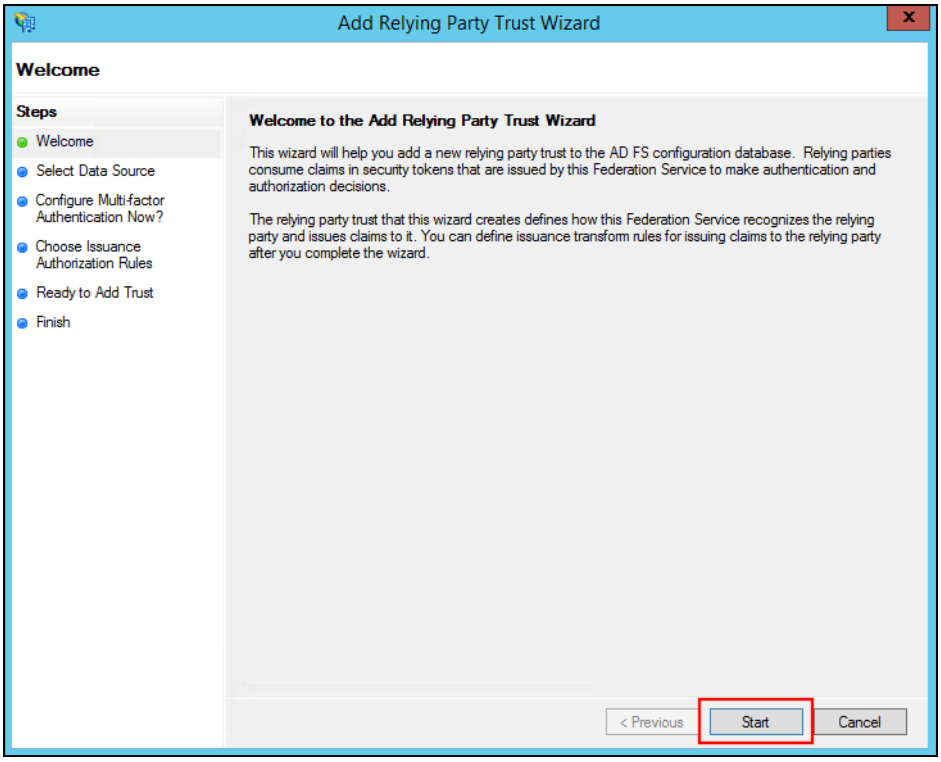
4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



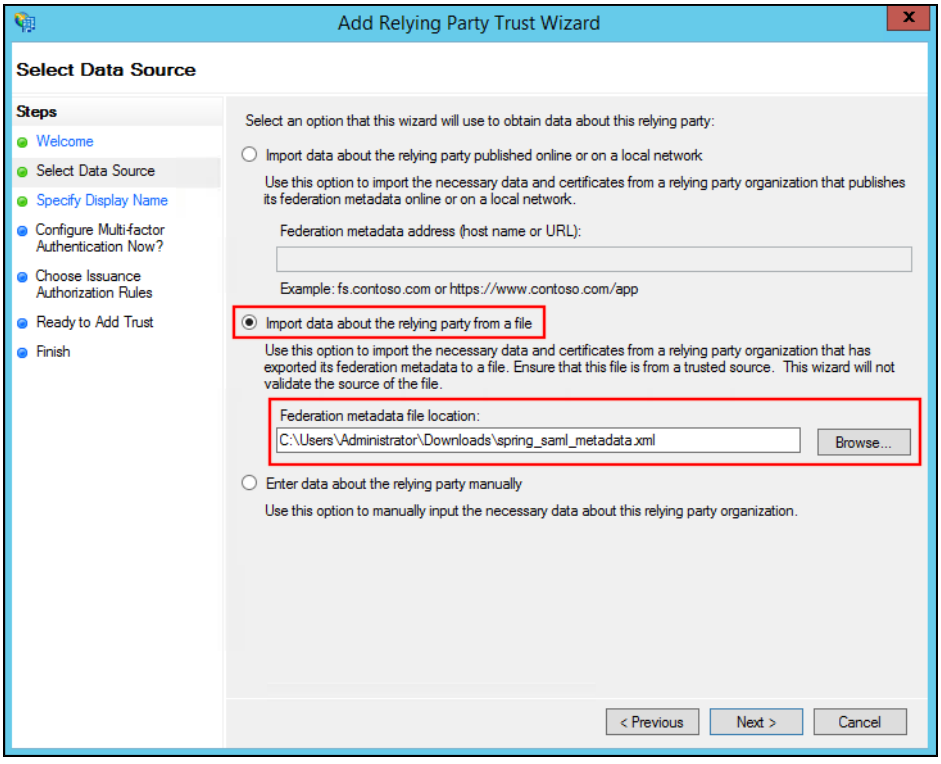
5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

Set Up SAML in Active Directory Federation Services

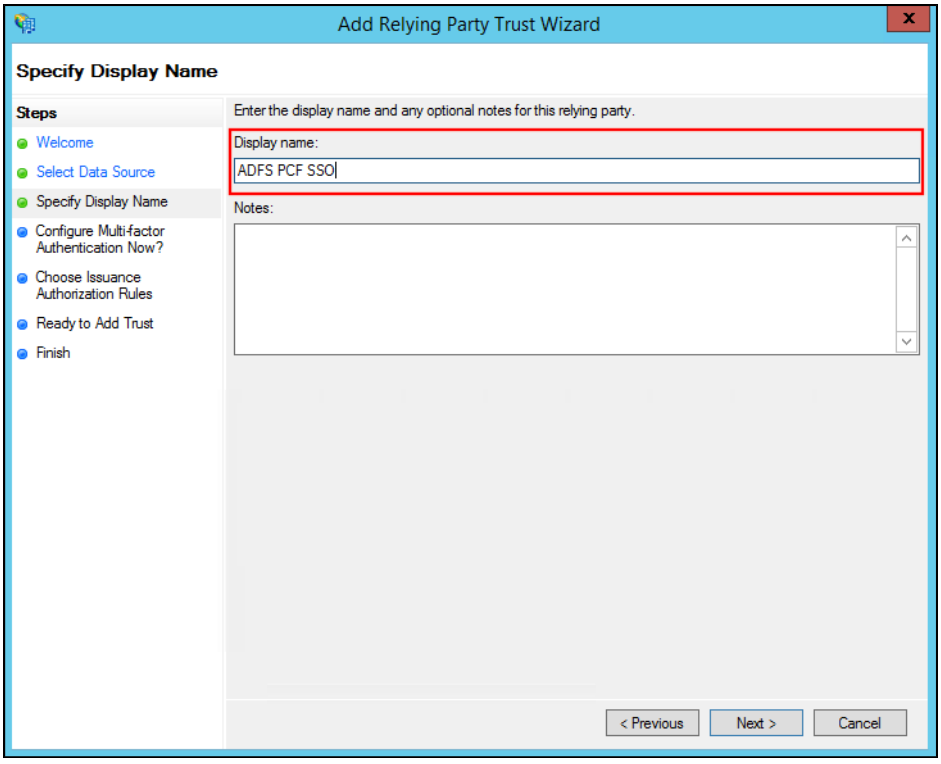
1. Open the **AD FS Management** console.
2. Click **Add Relying Party Trust...** in the Actions pane.
3. On the Welcome step, click **Start**.



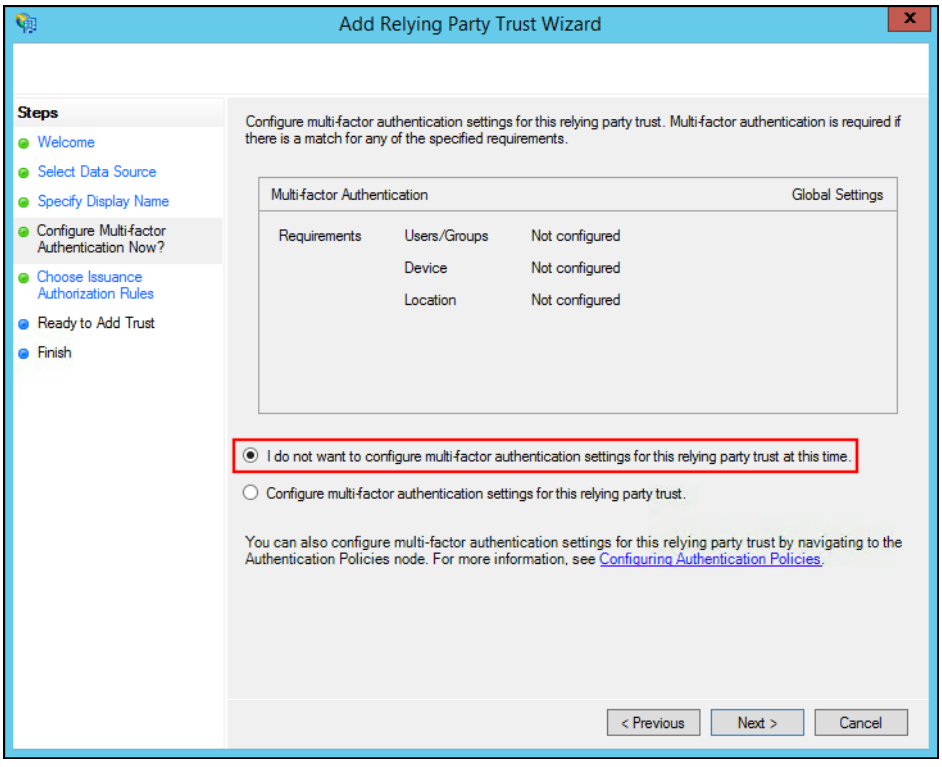
4. Select **Import data about the relying party from a file**, enter the path to the downloaded service provider metadata, and click **Next**.



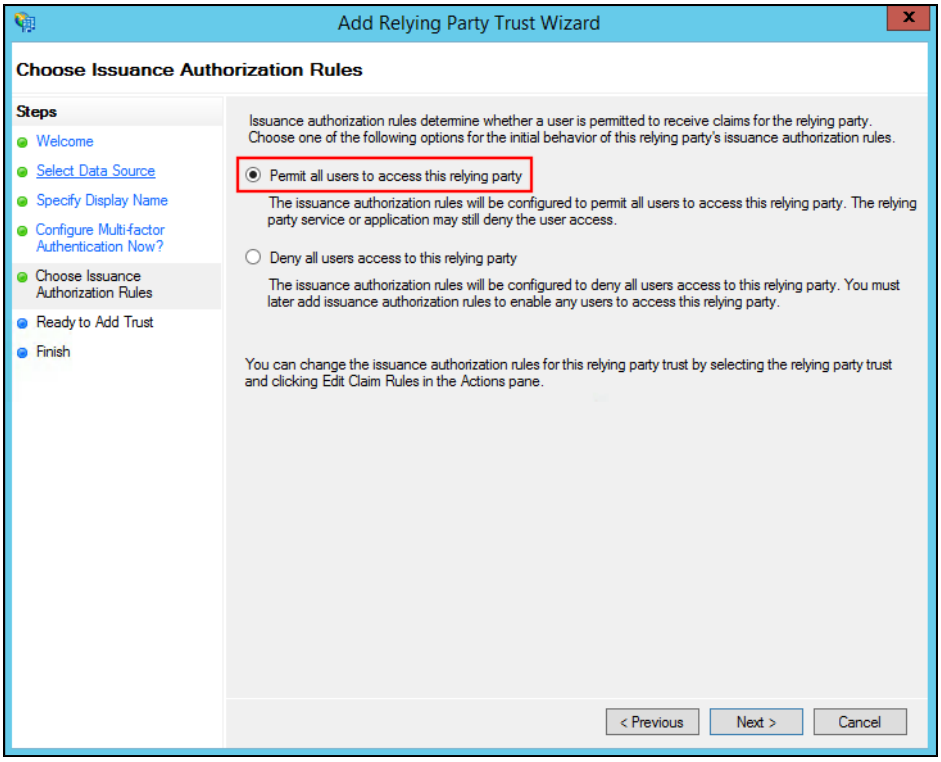
5. Enter a name for **Display name** and click **Next**.



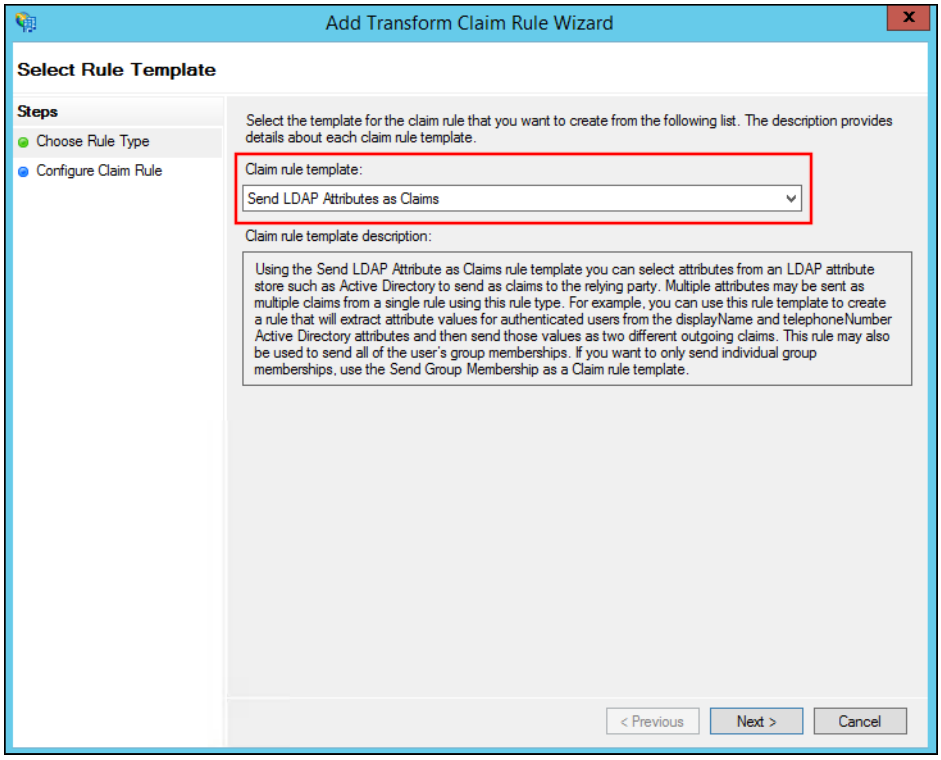
6. Leave the default multi-factor authentication selection and click **Next**.



7. Select **Permit all users to access this relying party** and click **Next**.

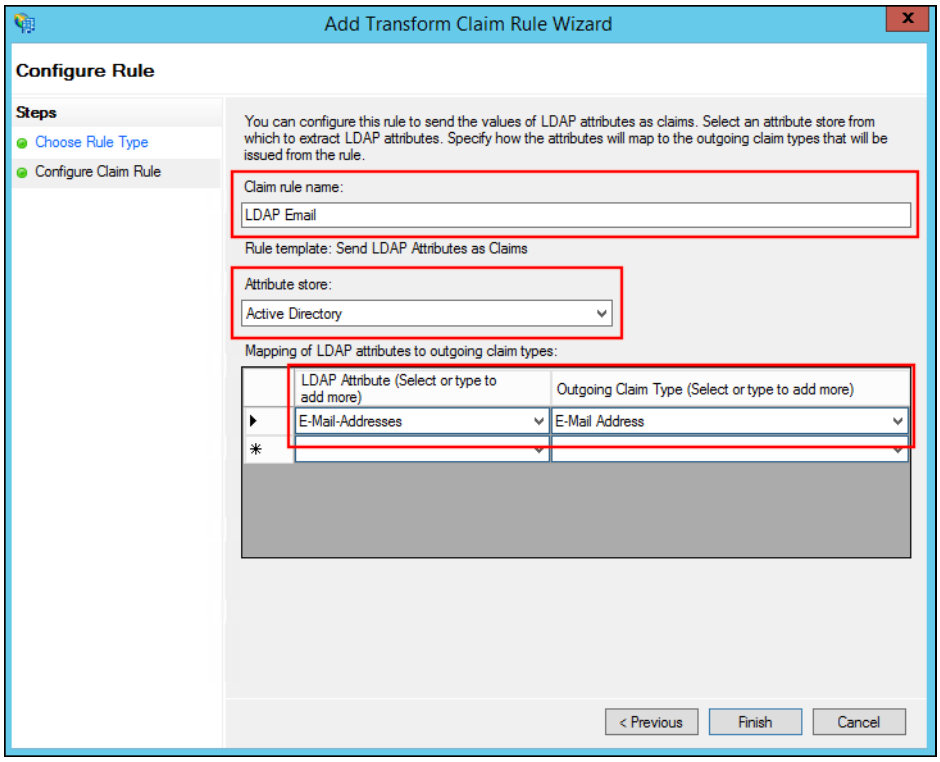


8. Review your settings and click **Next**.
9. Click **Close** to finish the wizard.
10. The claim rule editor should open by default. If it does not, select your Relying Party Trust and click **Edit Claim Rules...** in the Actions pane.
11. Create two claim rules by following these steps:
    a. Click **Add Rule**.
    b. Select **Send LDAP Attributes as Claims** for **Claim rule template** and click **Next**.



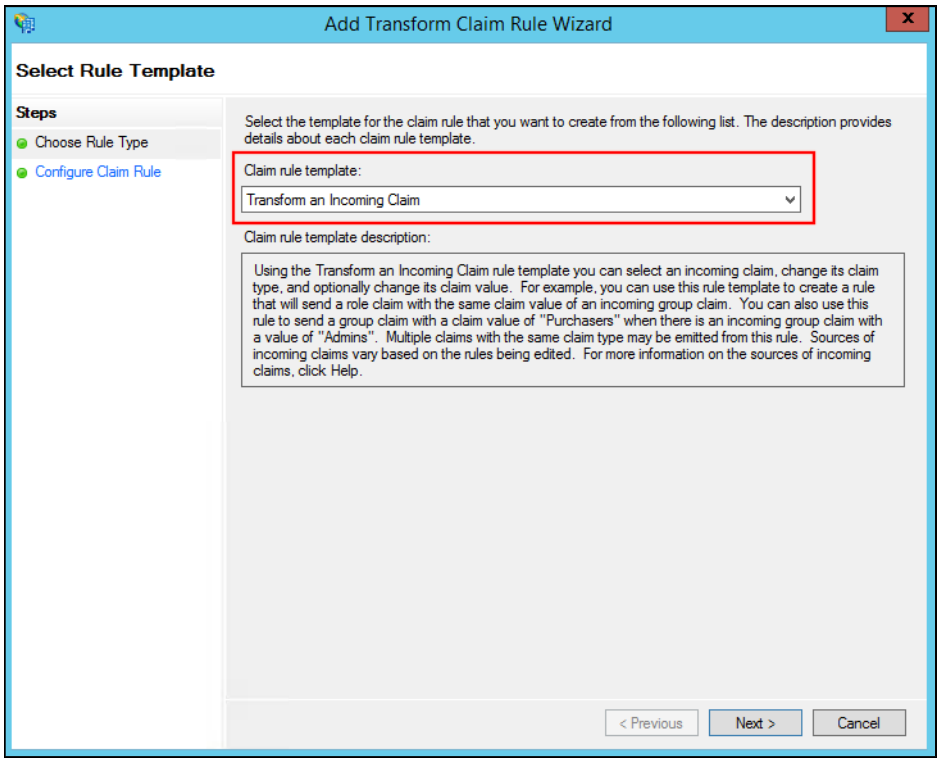
    c. Enter a **Claim rule name**.
    d. Select **Active Directory** for **Attribute store**.
    e. Select **E-Mail-Addresses** for **LDAP Attribute** and select **E-mail Address** for **Outgoing Claim Type**.

f. Click **Finish**.



g. Click **Add Rule**.

h. Select **Transform an Incoming Claim** for **Claim rule template** and click **Next**.



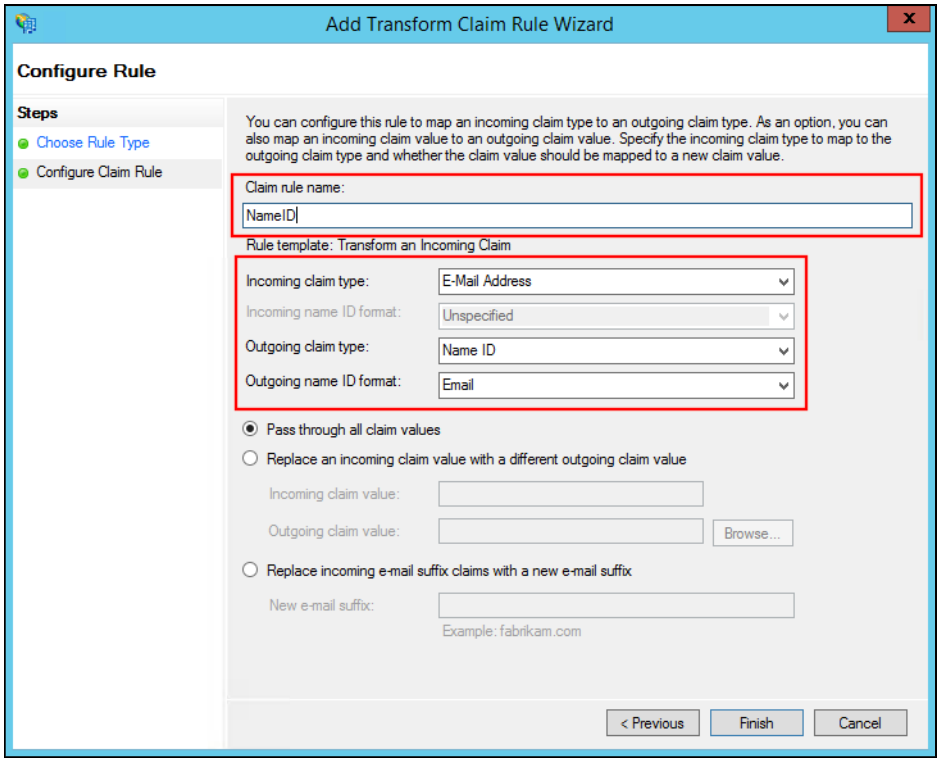
i. Enter a **Claim rule name**.

j. Select **E-Mail Address** for **Incoming claim type**.

k. Select **Name ID** for **Outgoing claim type**

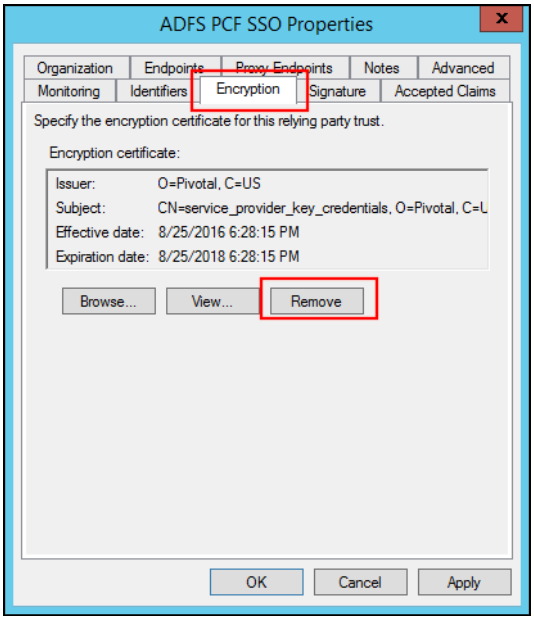
l. Select **Email** for **Outgoing name ID format**.

m. Click **Finish**.

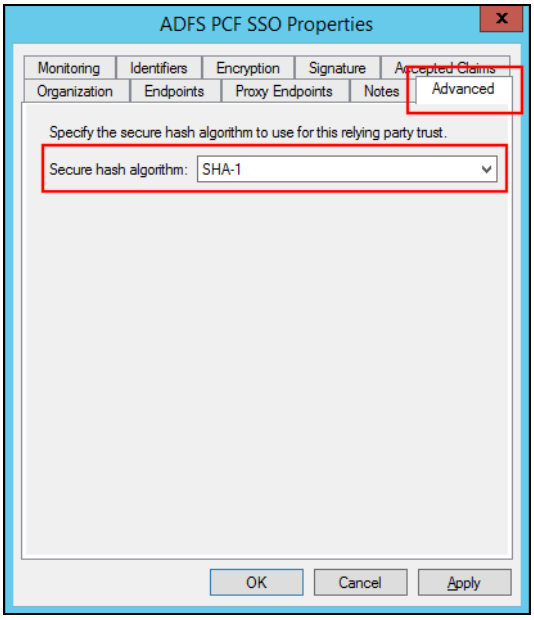


12. Double-click on the new Relying Party Trust to open the properties.

13. Select the **Encryption** tab and click **Remove** to remove the encryption certificate.



14. Select the **Advanced** tab and select the SHA algorithm for the **Secure hash algorithm** that matches the SHA Algorithm configured for PCF Elastic Runtime.



15. (Optional) If you are using a self-signed certificate, disable CRL checks by following these steps:

    a. Open **Windows Powershell** as an Administrator.
    b. Execute the following command:

    ```
    > set-ADFSRelyingPartyTrust -TargetName "< Relying Party Trust >" -SigningCertificateRevocationCheck None
    ```

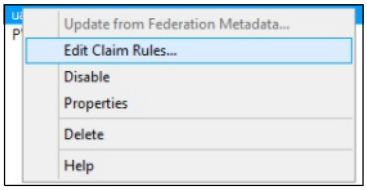
16. (Optional) If you are using a self-signed certificate, add it to the ADFS trust store. Obtain the Ops Manager certificate from `https://OPS_MANAGER_IP/api/v0/security/root_ca_certificate` and add this CA certificate to the ADFS trust store, so ADFS can trust the “Service Provider Key Certificate” certificate signed by OpsManager ROOT CA.

    Note: Prior to PCF v1.10, steps 13 and 14 are required as all PCF components (including SSO tile) have certificates are signed by an internal CA. In PCF v1.10+, customers can upload their own CA certificate to PCF.

17. (Optional) To specify any application or group attributes that you want to map to users in the ID token, click **Edit Claim Rules…** and configure **Send LDAP Attributes as Claims**. For more information, see the next section.

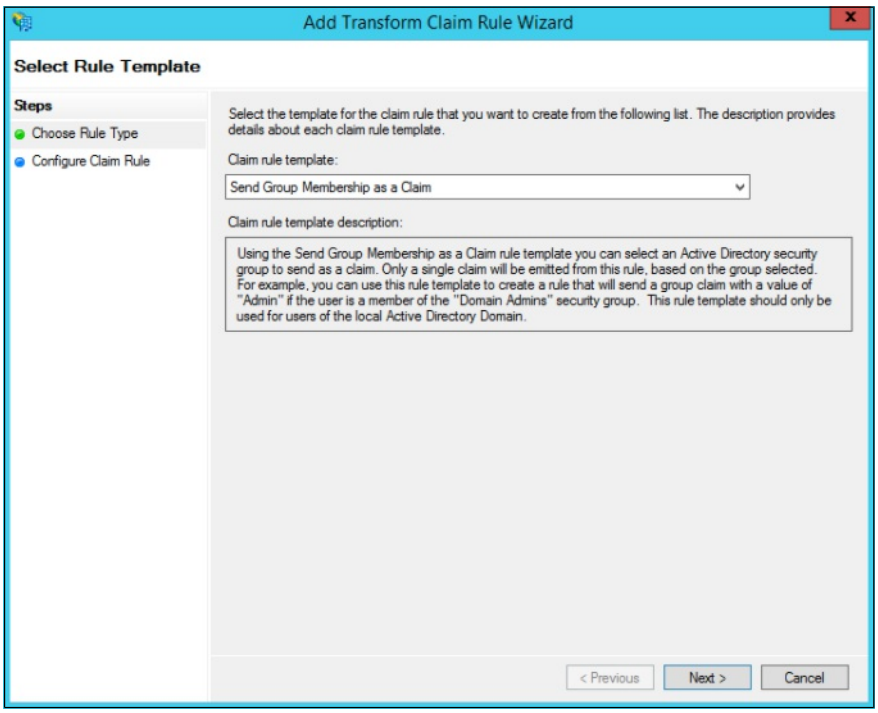
Setting Up Groups in SAML from ADFS

1. Right-click your **Relying Party Trust** and select **Edit Claim Rules…**.

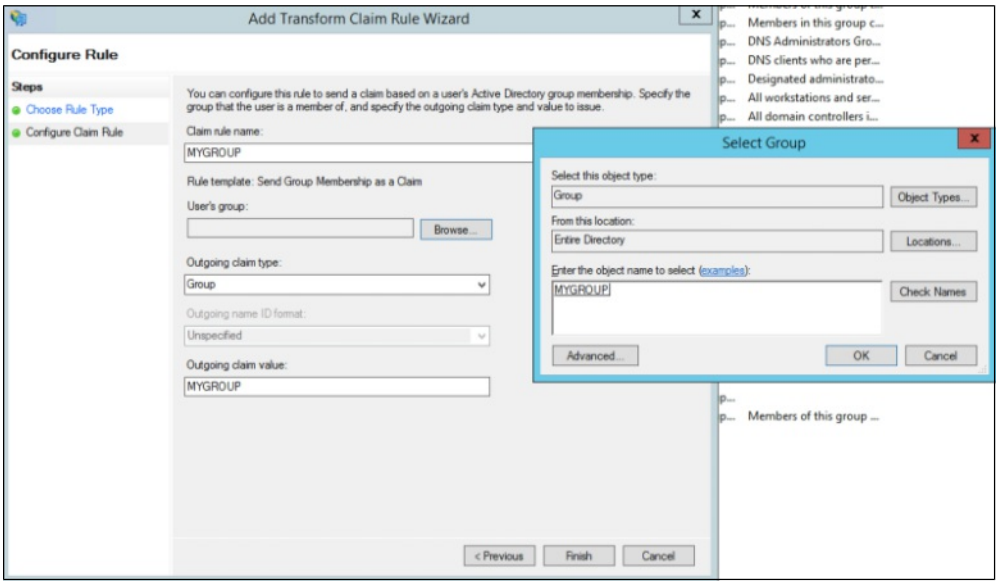


2. Select **Add Rule**.

3. Select **Send Group Membership as a Claim** and click **Next**.

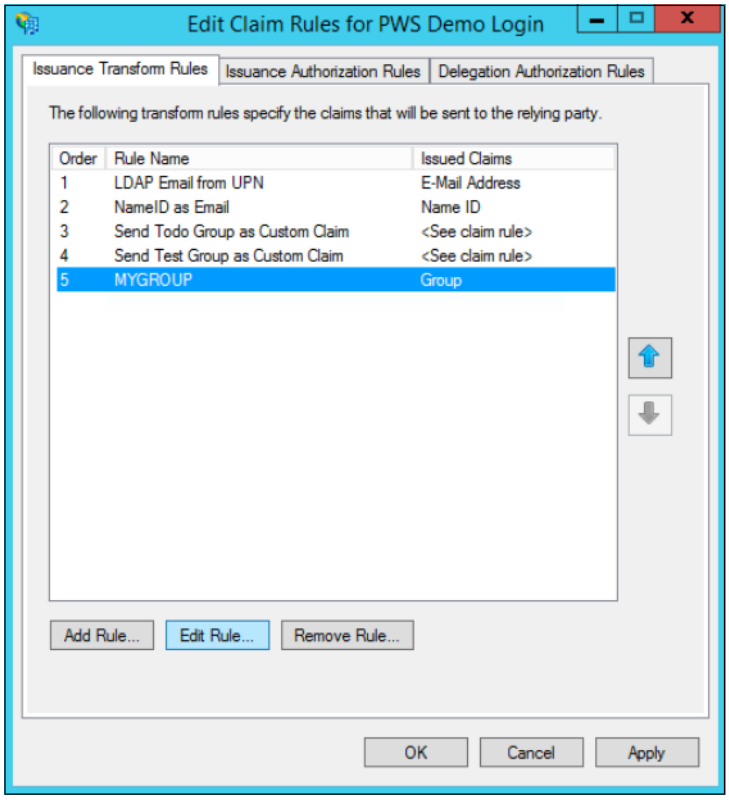


4. Enter the **Claim rule name**.
5. Click **Browse** to select your **User's group**.
6. Select **Group** as your **Outgoing claim type**.
7. Set your **Outgoing claim value** to match your group's name.

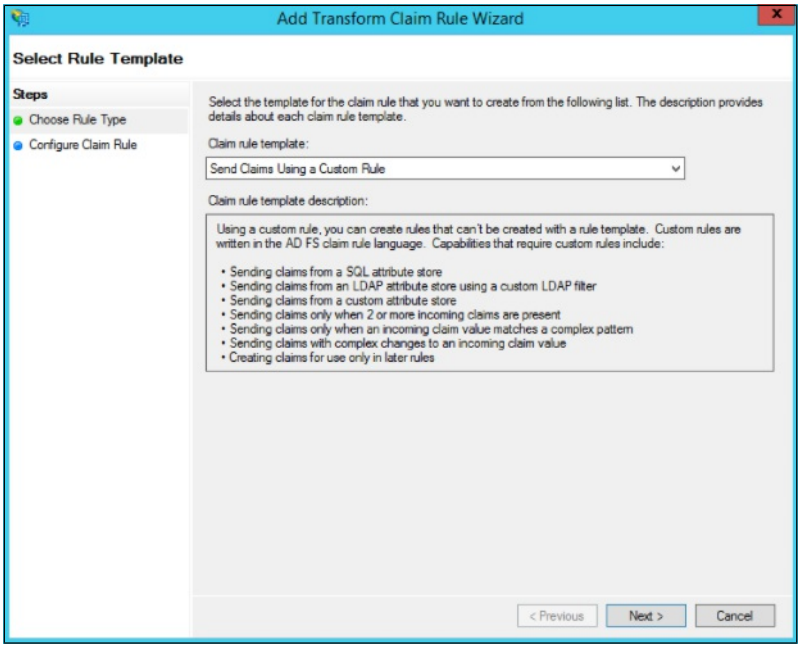


8. Click **Finish**.
9. To save these configurations and use the default SAML assertion of `http://schemas.xmlsoap.org/claims/Group`, click **OK**. If you want to pass the claims assertion as a custom value `"groups"` in the SAML assertion, continue to the procedure below.

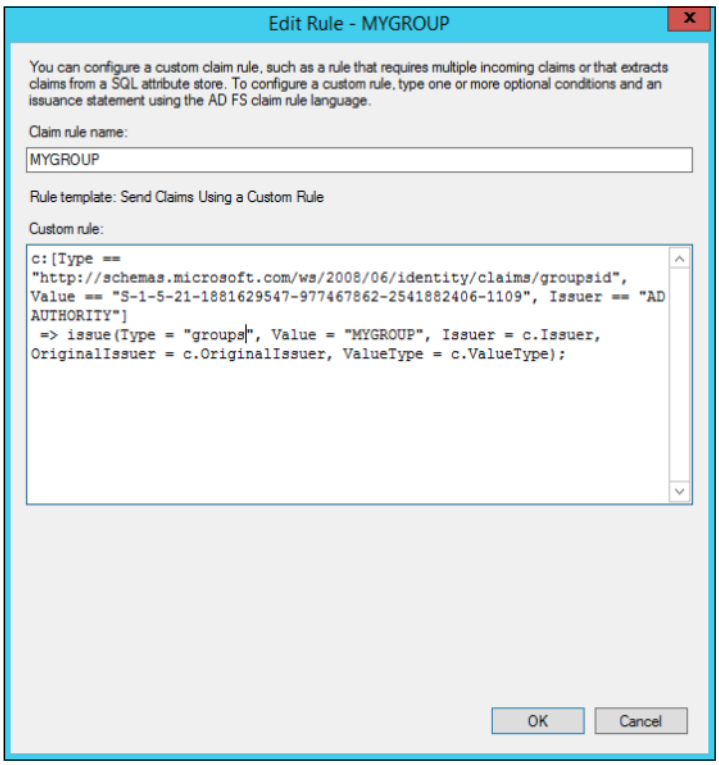
Create Custom Value "groups"



1. Select your newly created rule and click **Edit Rule**.
2. Click **View Rule Language**.
3. Copy the text in the **Claim rule language** field to a notepad or other location. You need this text for the next steps.
4. Exit the **Edit Rule** menu. Select the rule you just added and click **Remove Rule**.
5. Click **Add Rule**.



6. Select **Send Claims Using a Custom Rule**.
7. Paste in the text you previously copied in step 3 from the removed rule. Edit the `Type` so that it only says `"groups"`.



8. Click **OK** to finish making your changes and save the changes you made.

Configure a Single Sign-On Service Provider

This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Download Identity Provider Metadata

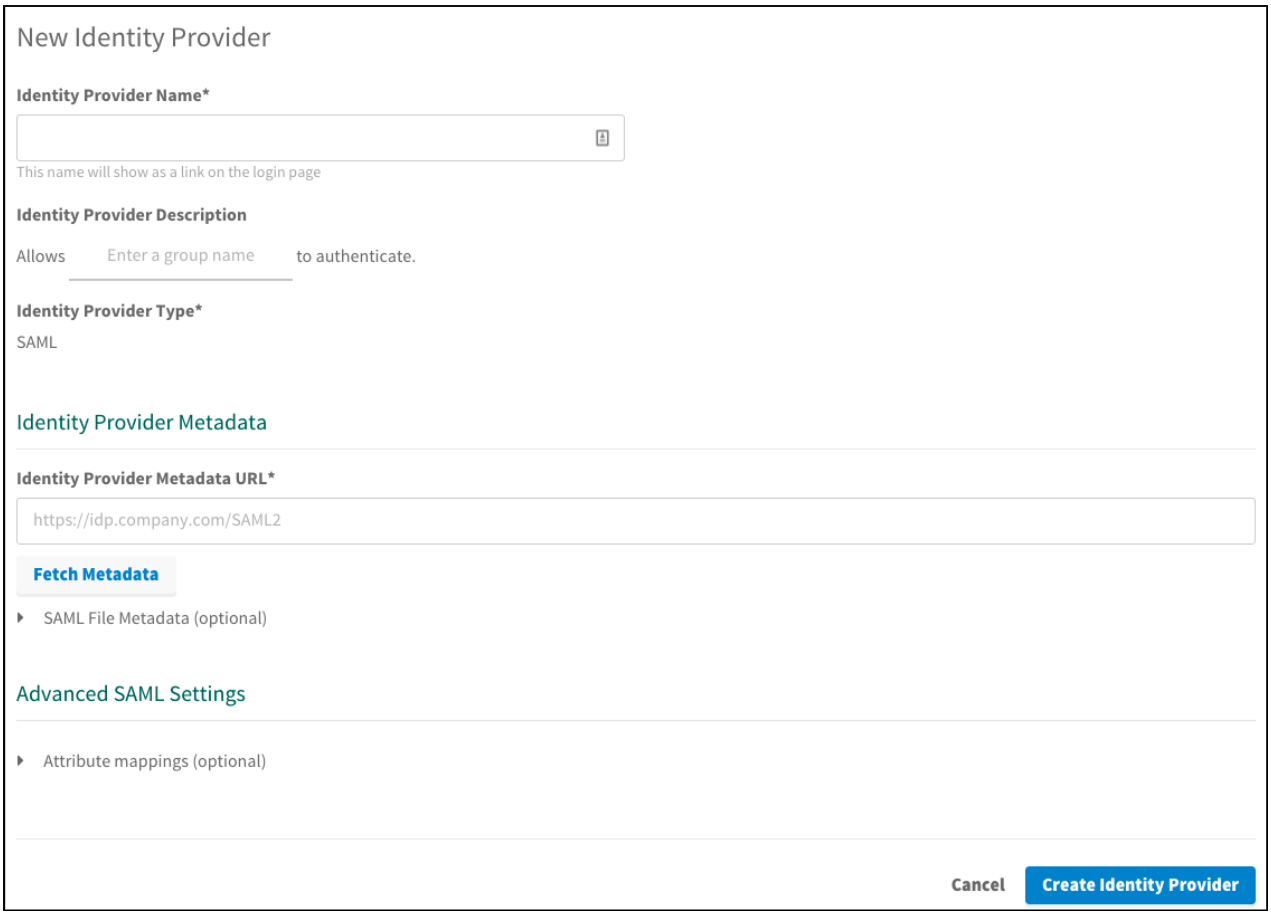
1. Download the metadata from your Active Directory Federation Services server at the following URL:
   `https://YOUR-ADFS-HOSTNAME/federationmetadata/2007-06/federationmetadata.xml`

Setting up SAML

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **New Identity Provider** to create a new identity provider.



4. To create a new identity provider, perform the following steps:

   a. Enter an identity provider name in **Identity Provider Name**.

b. (Optional) Enter a description in **Identity Provider Description**.
c. Click **SAML File Metadata (optional)**, then click **Upload Identity Provider Metadata** to upload your metadata XML.
d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

6. Click **Resource Permissions**.

7. Click **New Permissions Mapping** and perform the following steps:

   a. Enter a **Group Name**.
   b. For **Select Permissions**, select the permissions to grant to the members of the group from the external identity provider.

8. Navigate to the identity provider list.

9. Click **Group Whitelist** and enter the group names from the external identity provider that should be propagated in the ID token.

Create Attribute Mappings for SAML Groups

Under **User Attributes**, map the User Schema Attribute of `"external_groups"` to the Attribute Name value of `"groups"`. If you did not perform the steps to customize the SAML assertion value, use `"http://schemas.xmlsoap.org/claims/Group"` as the Attribute Name instead.

An attribute mapping with a customized SAML assertion value looks like this:

An attribute mapping with a non-customized SAML assertion value looks like this:

Groups now show up from the SAML assertion as claims. You can pull these values from the user's stored custom attributes using the `roles` scope on the ID token or through the userinfo endpoint, or map these to permissions using Resource Permissions mappings. For more information, see the *Create or Edit Resource Permissions* section of Manage Resources.

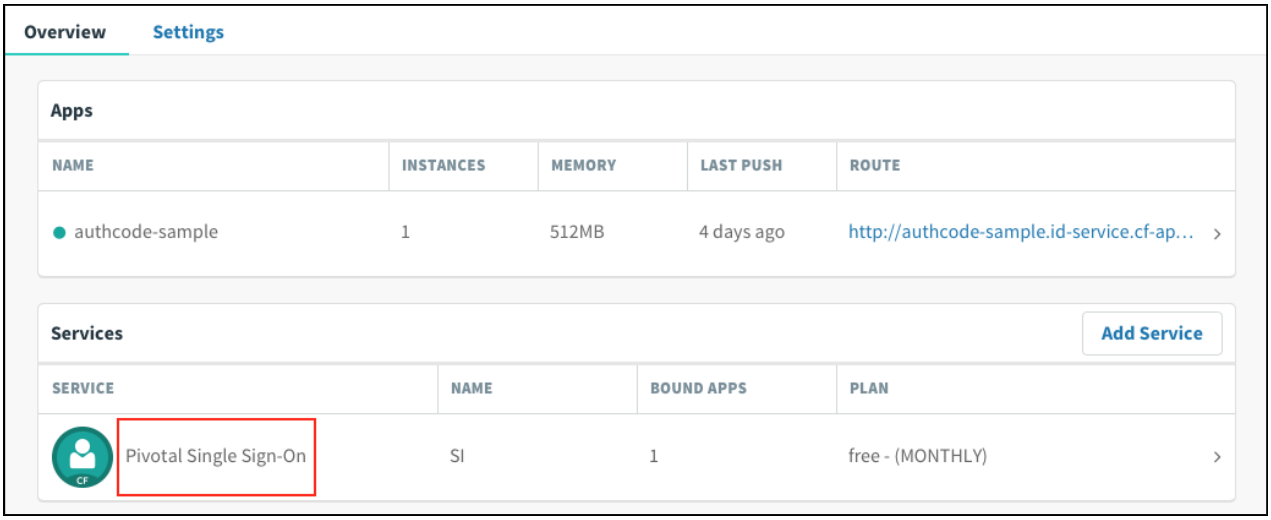
Testing

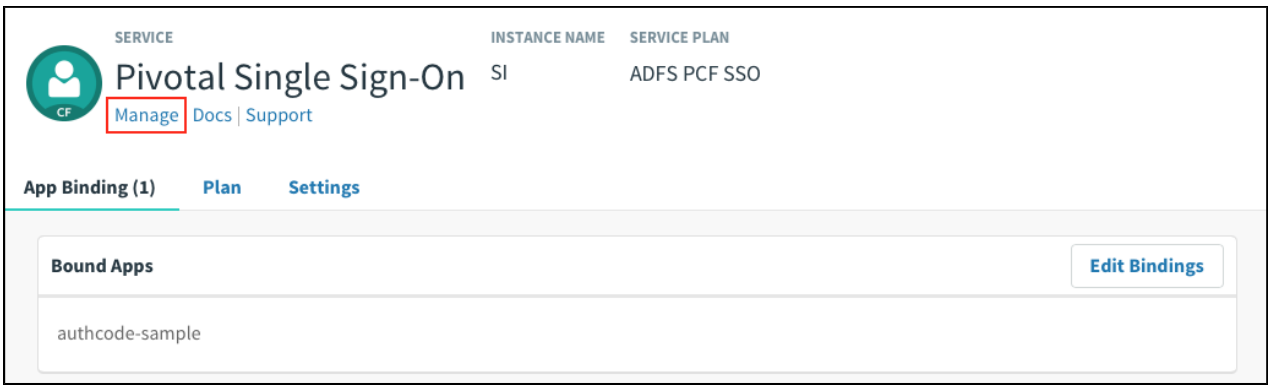
This topic describes how an administrator can test the connection between SSO and Active Directory Federation Services (AD FS). An administrator can test both service provider and identity provider connections.

Test Your Service Provider Connection

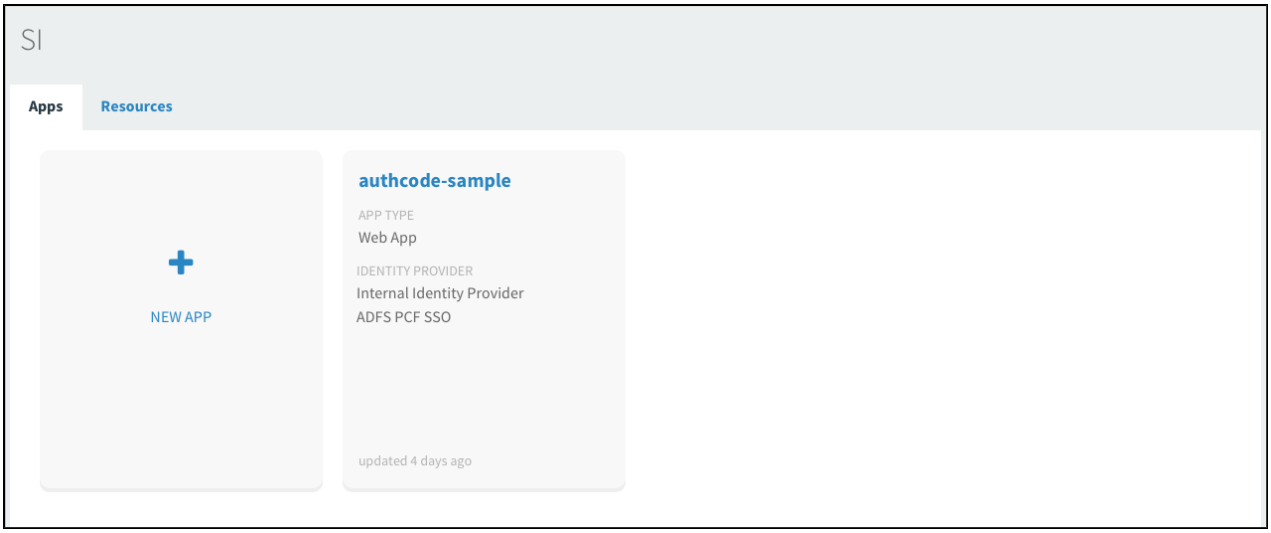
1. Log in to Apps Manager at https://apps.YOUR-SYSTEM-DOMAIN and navigate to the organization and space where your application is located.

2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Click on the service instance and click **Manage**.

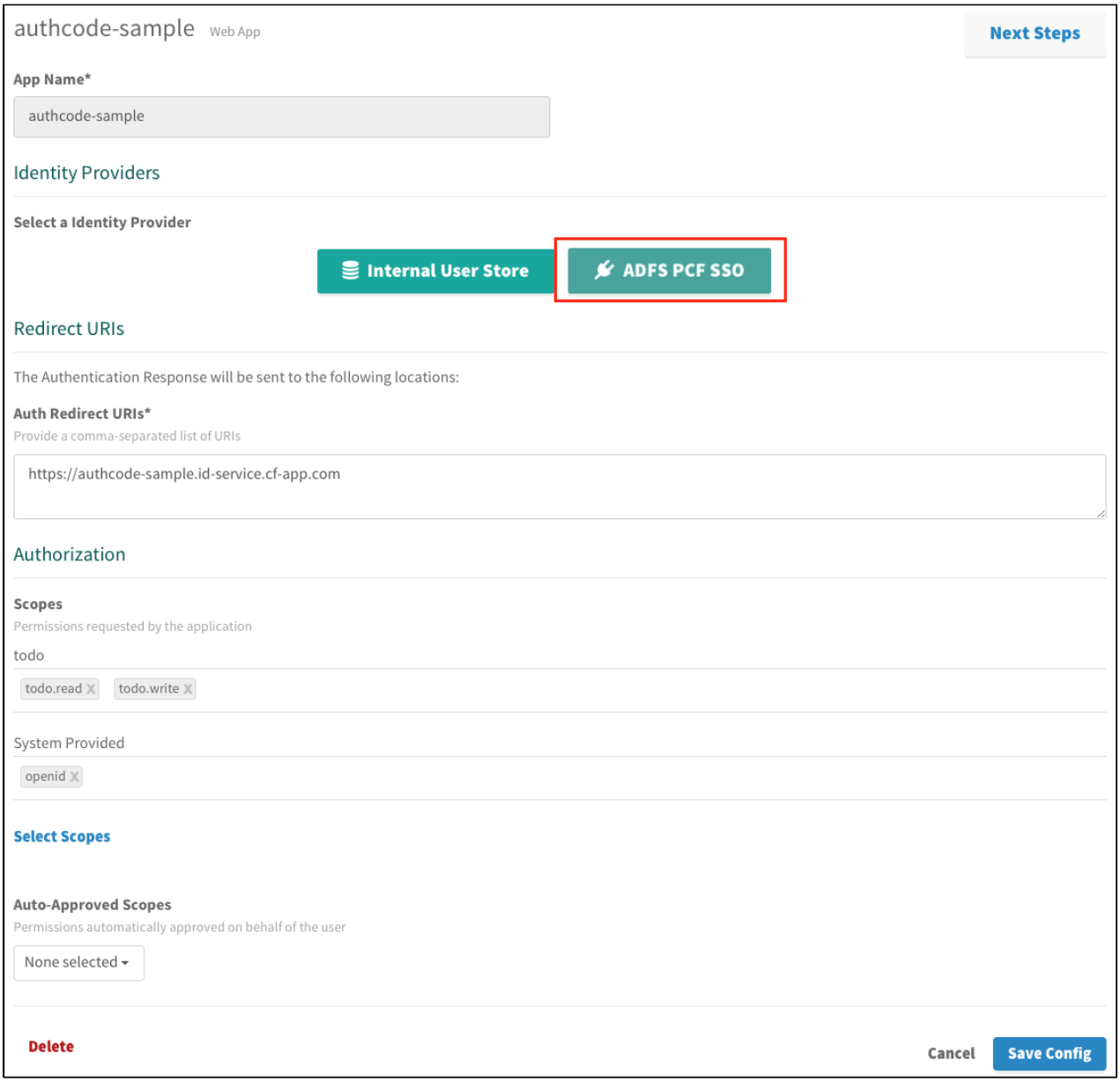




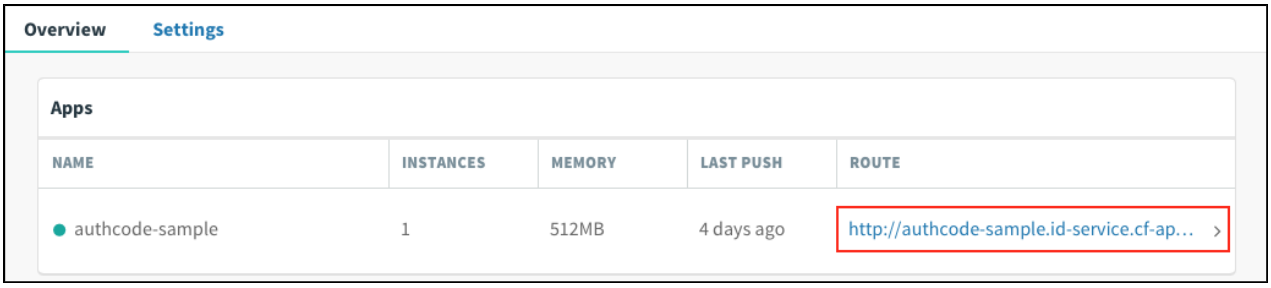
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the AD FS identity provider.



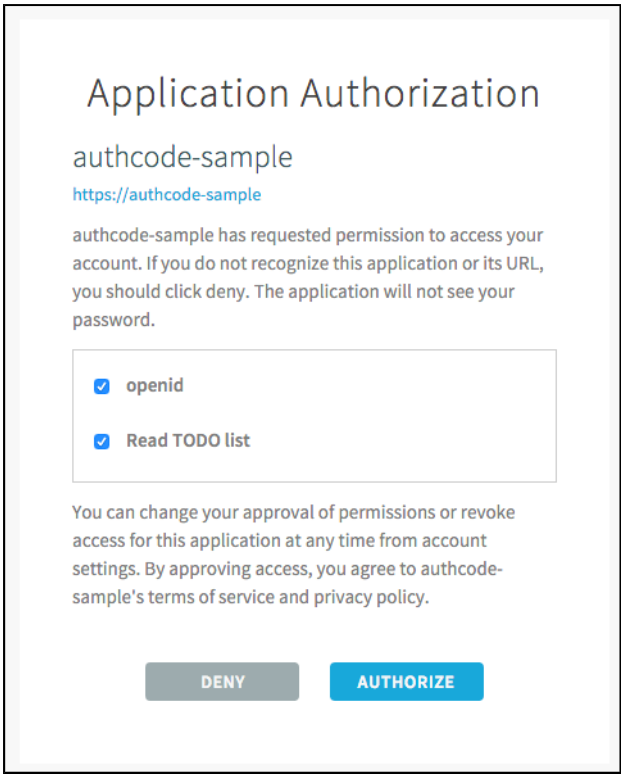
5. Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.



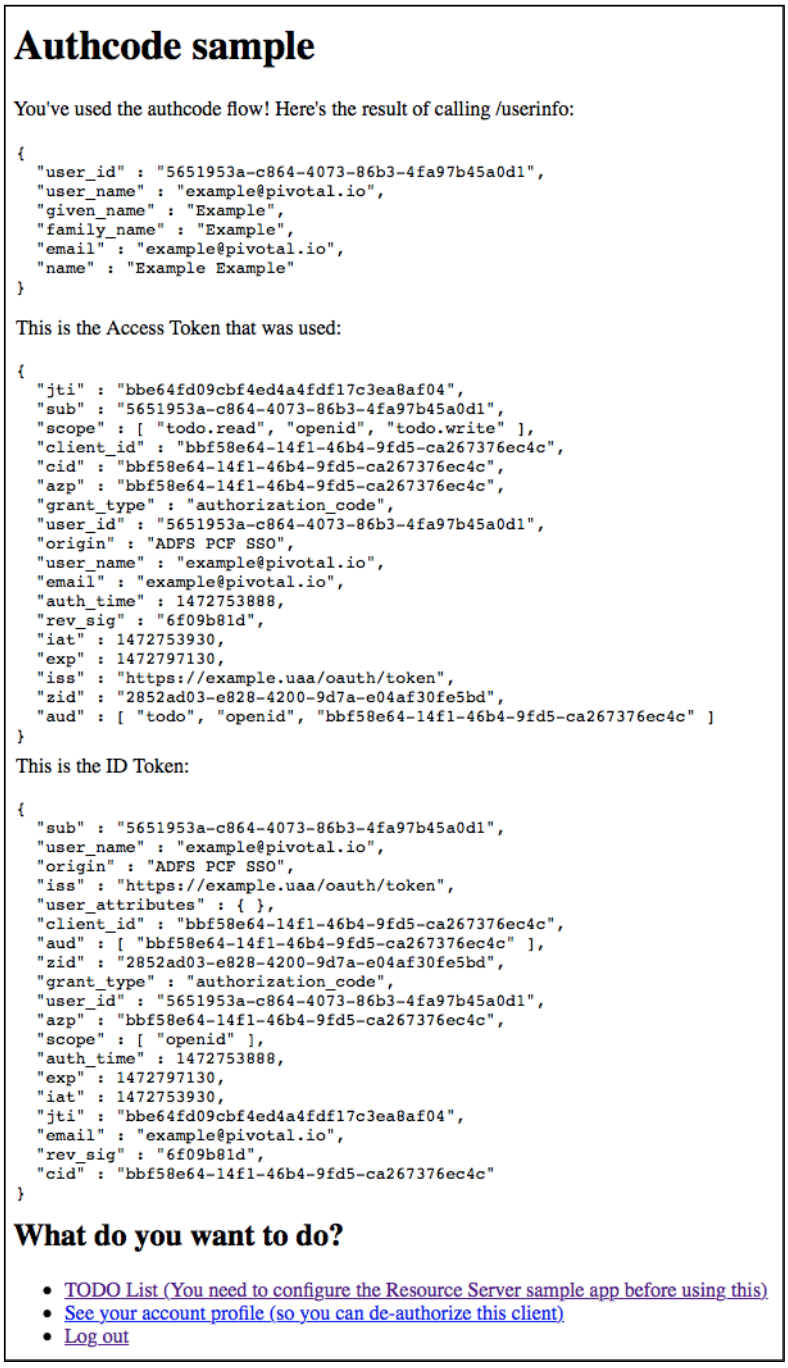
6. Click the link.

7. On the identity provider sign-in page, enter your credentials and click **Sign in**.

8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. The access token and ID token displays.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "user_name" : "example@pivotal.io",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "bbe64fd09cbf4ed4a4fdf17c3ea8af04",
  "sub" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "cid" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "azp" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "grant_type" : "authorization_code",
  "user_id" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "origin" : "ADFS PCF SSO",
  "user_name" : "example@pivotal.io",
  "email" : "example@pivotal.io",
  "auth_time" : 1472753888,
  "rev_sig" : "6f09b81d",
  "iat" : 1472753930,
  "exp" : 1472797130,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "2852ad03-e828-4200-9d7a-e04af30fe5bd",
  "aud" : [ "todo", "openid", "bbf58e64-14f1-46b4-9fd5-ca267376ec4c" ]
}
```


This is the ID Token:


```
{
  "sub" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "user_name" : "example@pivotal.io",
  "origin" : "ADFS PCF SSO",
  "iss" : "https://example.uaa/oauth/token",
  "user_attributes" : { },
  "client_id" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "aud" : [ "bbf58e64-14f1-46b4-9fd5-ca267376ec4c" ],
  "zid" : "2852ad03-e828-4200-9d7a-e04af30fe5bd",
  "grant_type" : "authorization_code",
  "user_id" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "azp" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "scope" : [ "openid" ],
  "auth_time" : 1472753888,
  "exp" : 1472797130,
  "iat" : 1472753930,
  "jti" : "bbe64fd09cbf4ed4a4fdf17c3ea8af04",
  "email" : "example@pivotal.io",
  "rev_sig" : "6f09b81d",
  "cid" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c"
}
```


What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

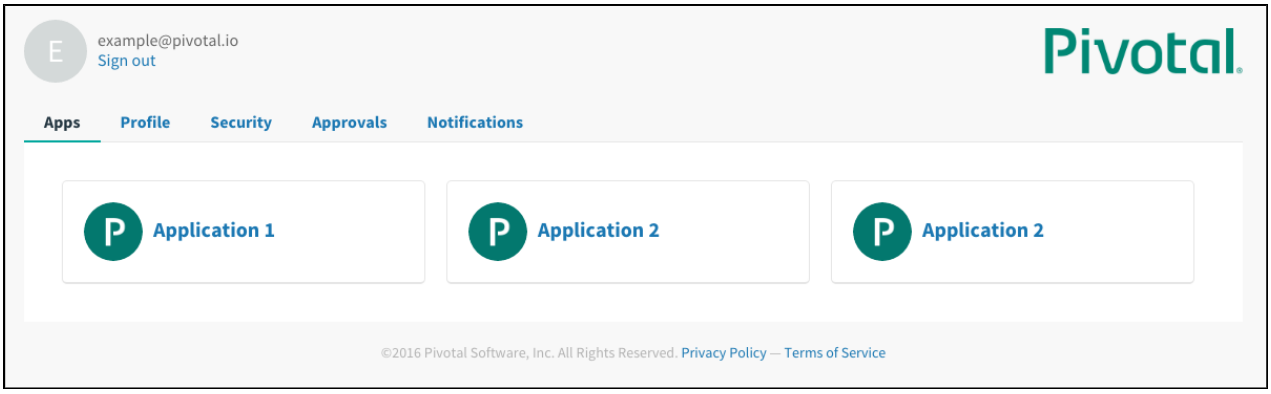
Test Your Identity Provider Connection

Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to AD FS.

2. Navigate to your application and click it.

3. You are redirected to the page that lists applications you have access to.

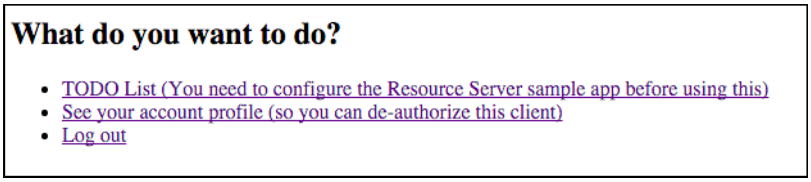


Test Your Single Sign-Off

Test single sign-off to ensure that when users log out of the application, they are logged out of AD FS as well.

1. Sign in to the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

2. Under “What do you want to do?”, click **Log out**.



3. You are logged out and redirected to the AD FS login page.

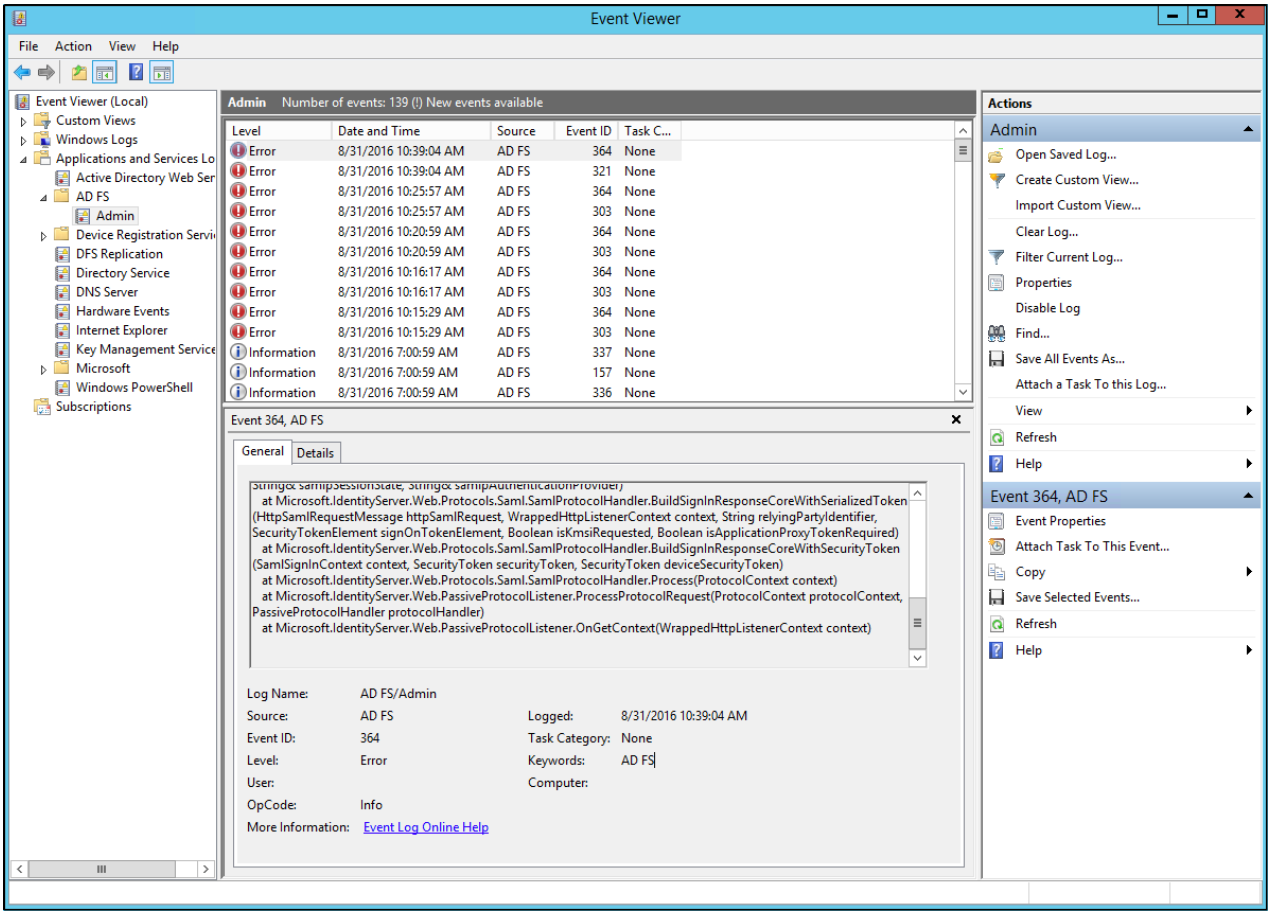
Troubleshooting

This topic describes how to resolve errors that arise when configuring a single sign-on partnership between Active Directory Federation Services and Pivotal Single Sign-On (SSO).

Event Viewer

1. Navigate to **Administrative Tools**.

2. Launch **Event Viewer**.



3. Examine any errors and its details to diagnose problems.

Azure Active Directory SAML Integration Guide Overview

This documentation introduces how to set up Azure Active Directory (Azure AD) with Security Assertion Markup Language (SAML) as the identity provider for the Single Sign-On service running on Pivotal Cloud Foundry (PCF).

Azure Active Directory (Azure AD) is Microsoft's multi-tenant cloud based directory and identity management service.

For how to set up Azure AD with Open ID Connect (OIDC) , see Azure Active Directory OIDC Integration Guide.

Prerequisites

To integrate Azure AD with Pivotal Cloud Foundry® (PCF), you need:

Pivotal

- PCF, version 1.7.0 or later.
- Single Sign-On, version 1.1.0 or later.

Azure Active Directory

- Azure Active Directory subscription.
- A user with admin privileges.

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic.

Azure AD Integration Guide

Configuring Azure AD with SSO

Complete both steps below to integrate your deployment with Azure AD and SSO.

1. Configure Azure AD as a SAML Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

- Testing
- Troubleshooting

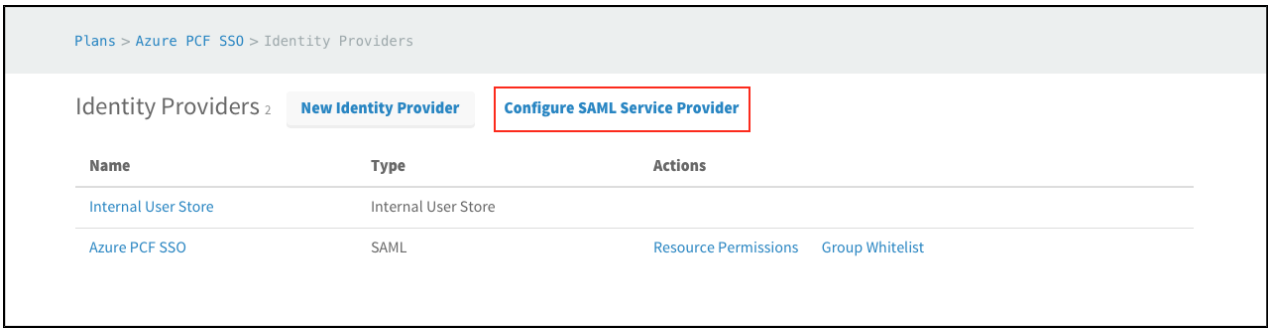
Configure Azure Active Directory as a SAML Identity Provider

This topic describes how to set up Azure Active Directory (AD) as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry® (PCF) and Azure AD.

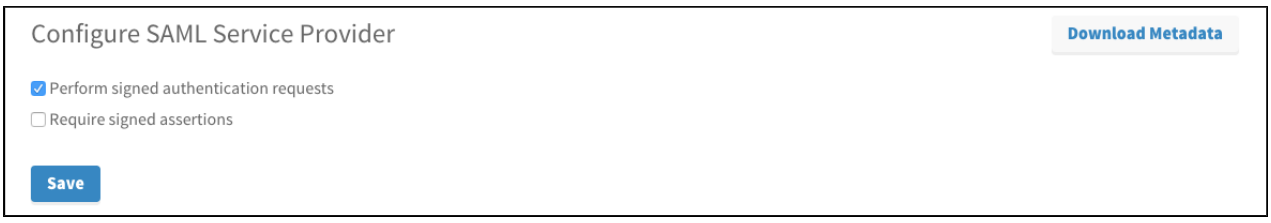
Set up SAML in PCF

1. Log in to the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **Configure SAML Service Provider**.



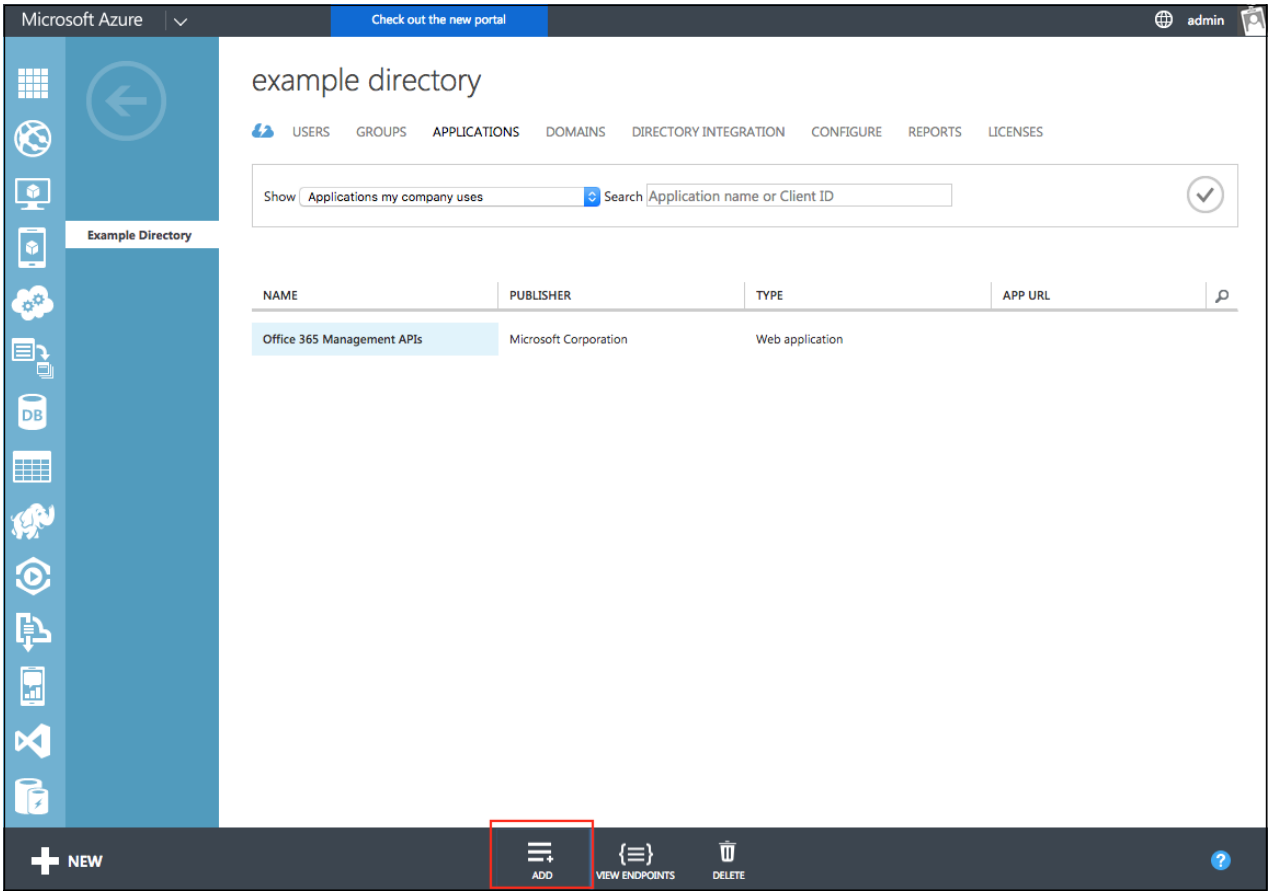
4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



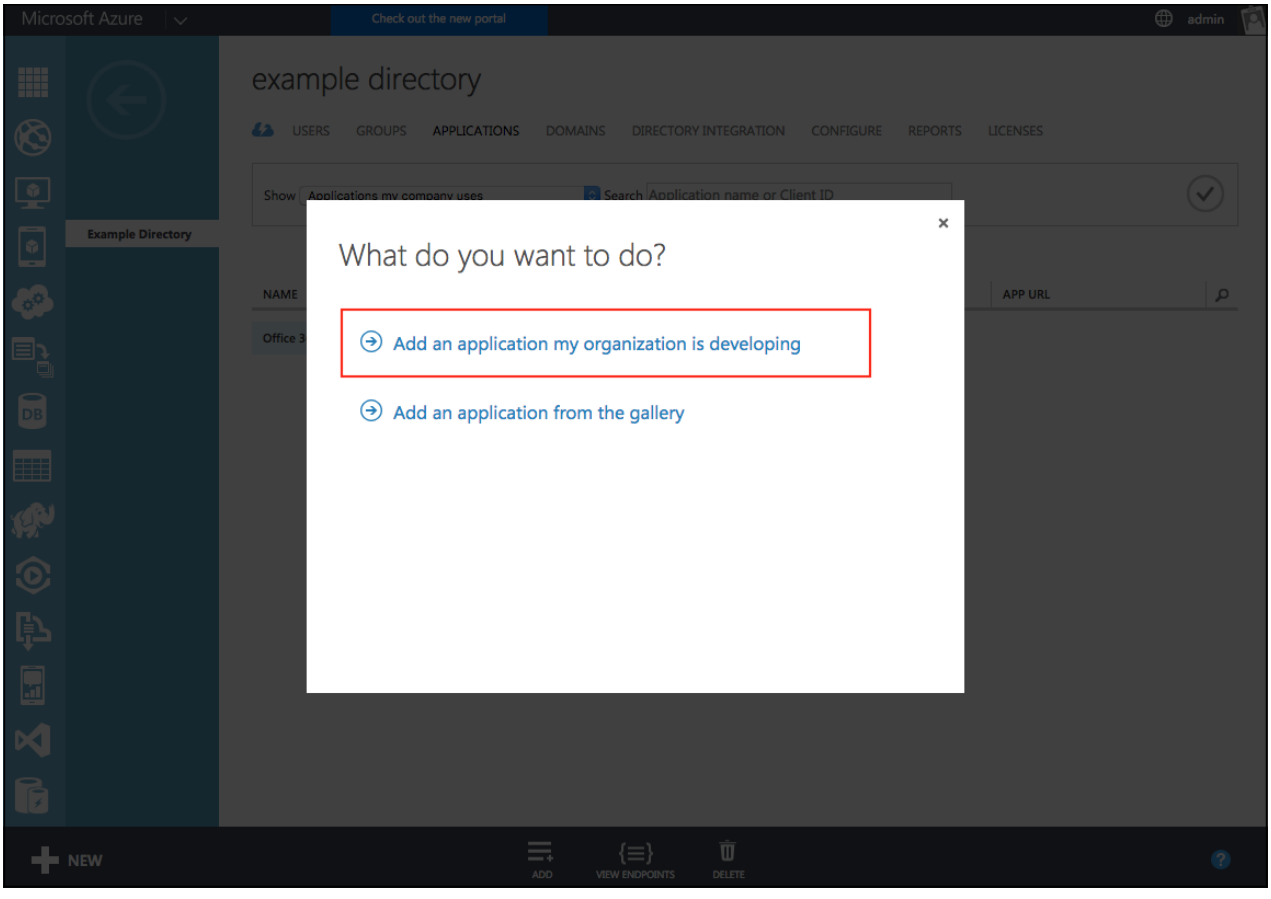
5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

Set up SAML in Azure Active Directory

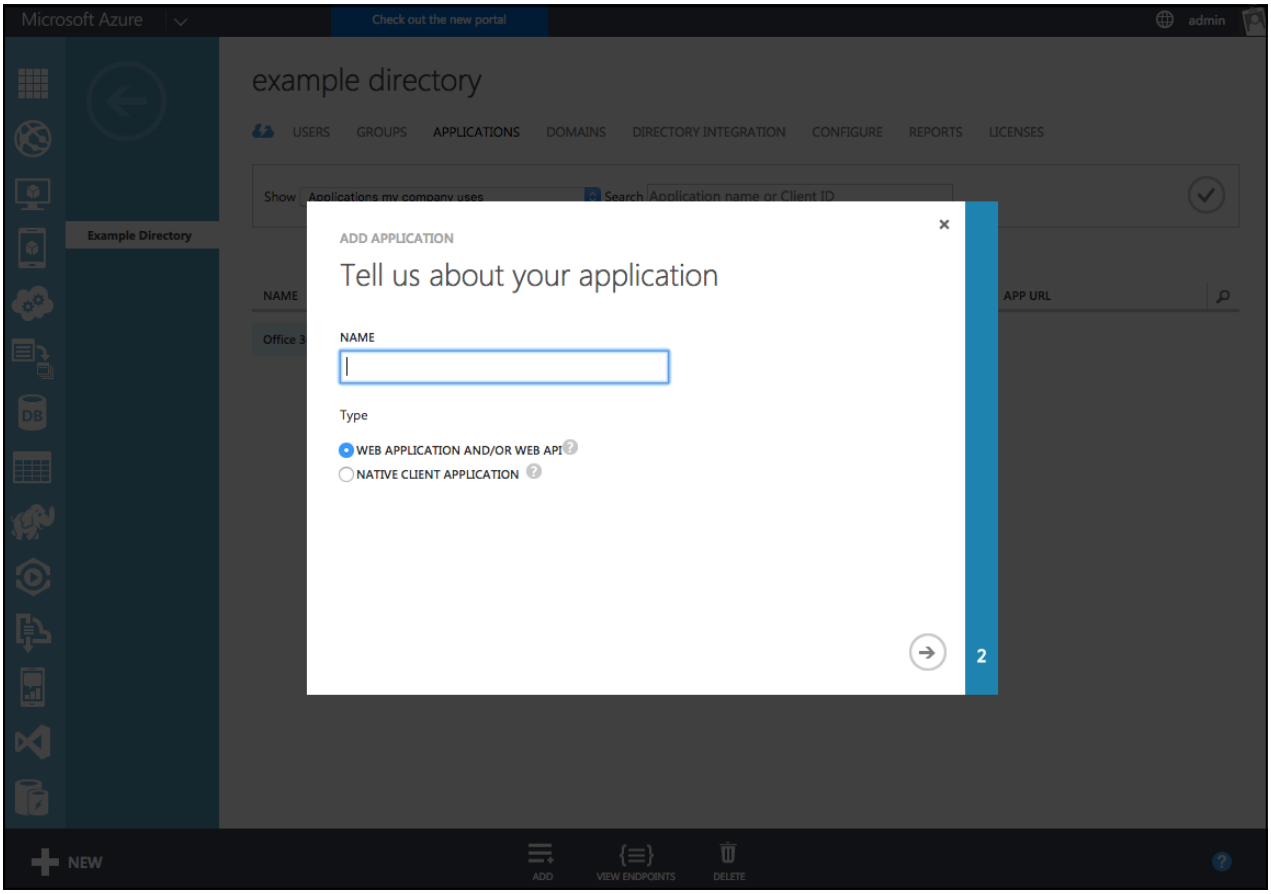
1. Sign into Azure AD at `https://manage.windowsazure.com` as an administrator.
2. Navigate to the applications dashboard by clicking on your directory and the **Applications** tab.
3. Click the **Add** button to add a new application.



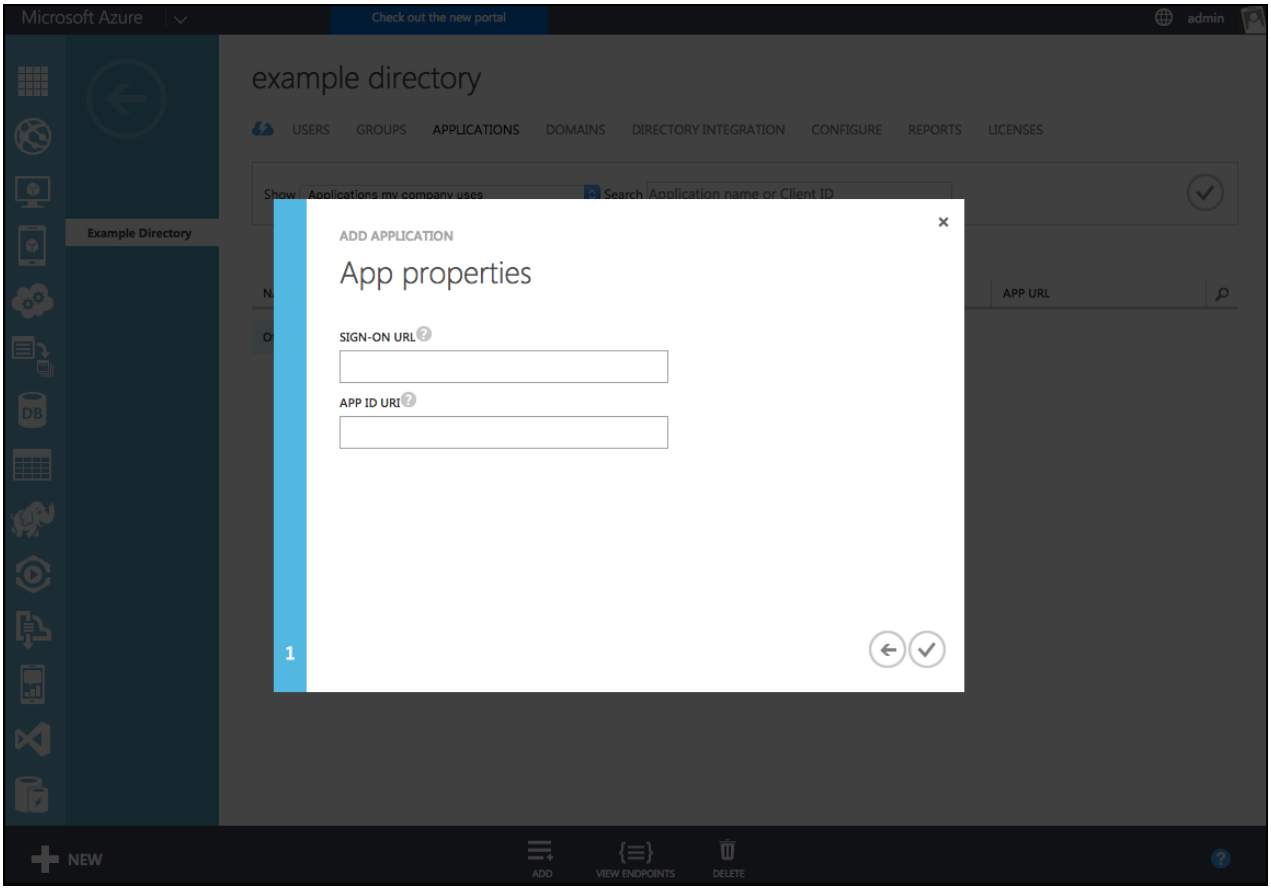
4. Select **Add an application my organization is developing**



5. Enter the **Name** and **Type** for the application.

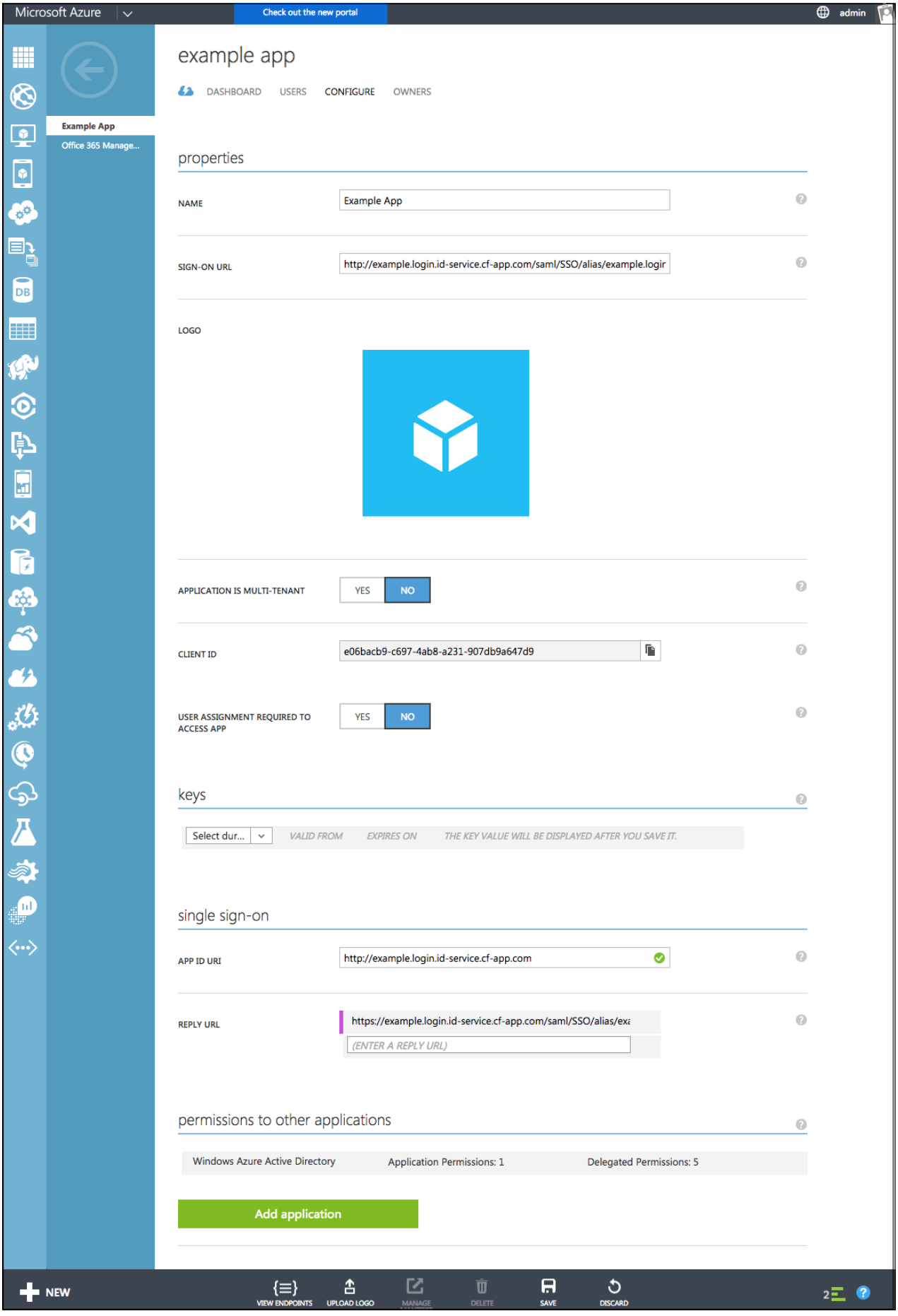


6. Enter the **Sign-On URL** and **App ID URI** for the application.



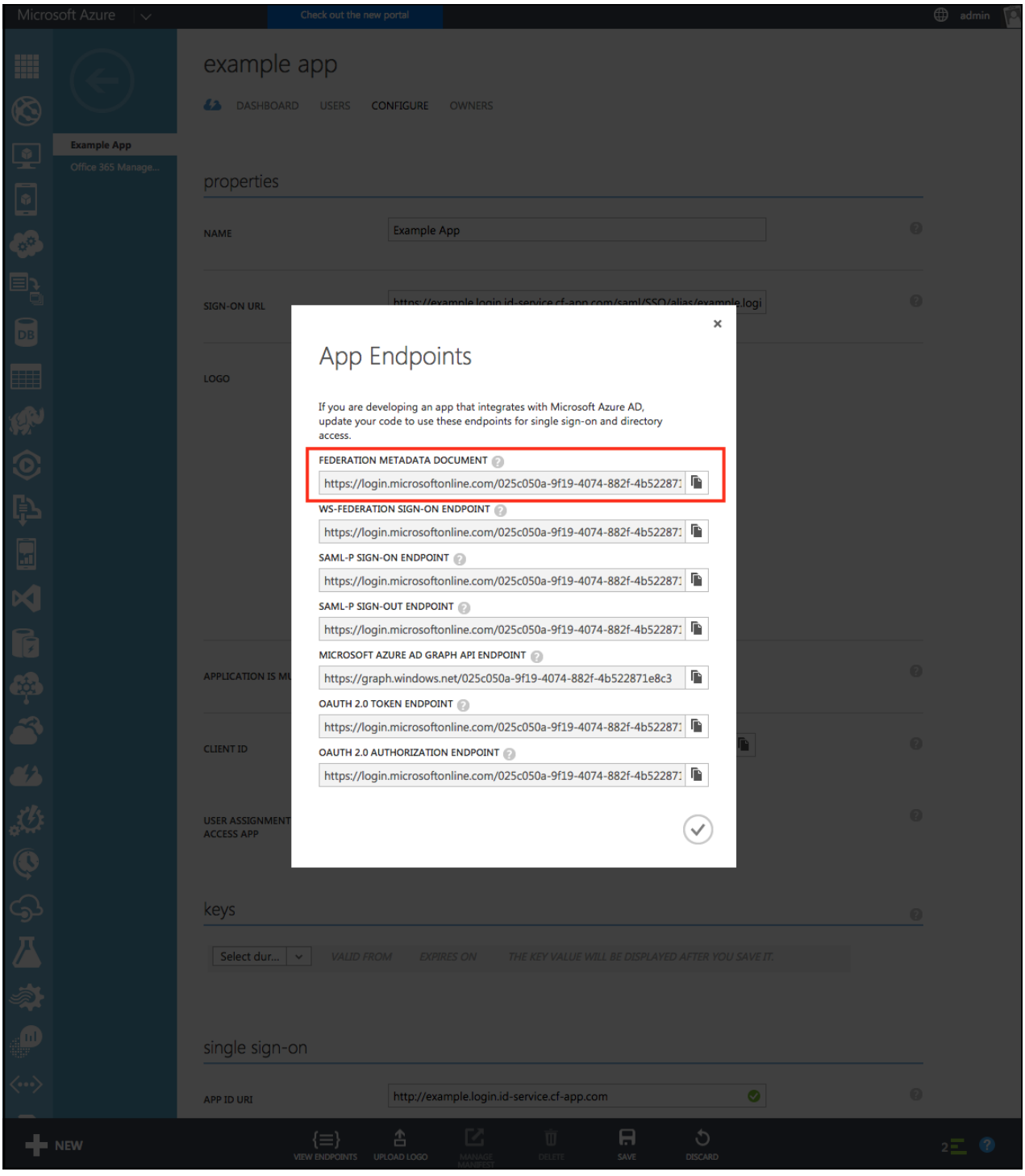
7. Click the application and configure the following properties:

a. Enter the application **Name**.
b. Enter the `AssertionConsumerService Location URL` from your downloaded service provider metadata into **Sign-On URL**. For example, `https://AUTH-DOMAIN/saml/SSO/alias/AUTH-DOMAIN`.
c. Configure the application **Logo**, **Application is Multi-Tenant** and **User Assignment Required to Access App** properties.
d. Enter your `Auth Domain URL` into **App ID URI**.
e. Enter the `AssertionConsumerService Location URL` from your downloaded service provider metadata into **Reply URL**.



8. Click the **Save** button.

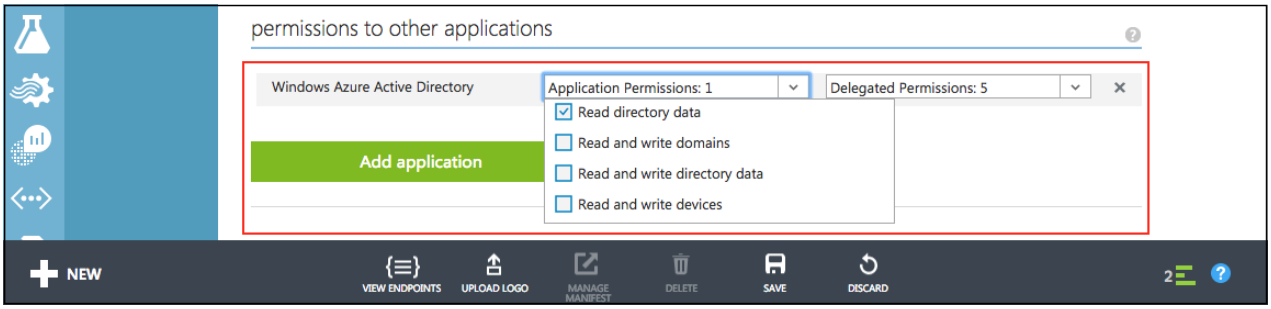
9. Click **View Endpoints** and download the **Federation Metadata Document**.



Set up Claims Mapping

1. To enable user attribute mappings, grant the application the following permissions to Windows Azure Active Directory:

    a. Read directory data.
    b. Read all groups.
    c. Read all users' full profiles or Read all users' basic profiles.

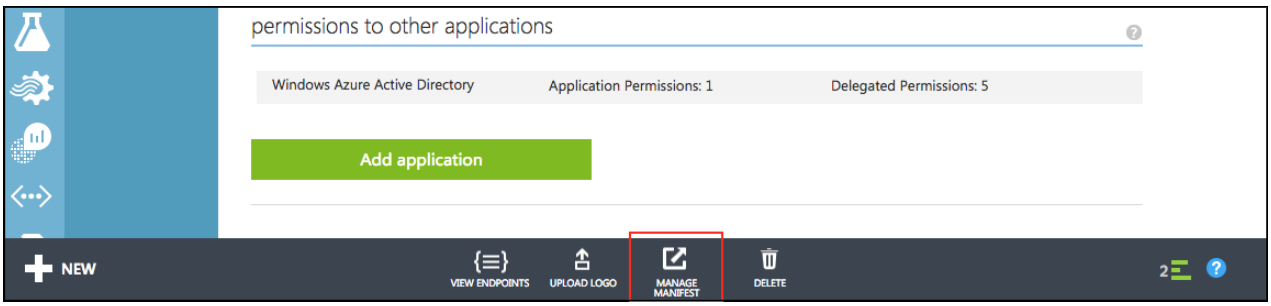


2. To pass group membership claims to the application, perform the following steps:

   a. Click **Manage Manifest**.
   b. Click **Download Manifest** followed by **Download manifest**.
   c. Locate `groupMembershipClaims` and set the value to either:

      - `SecurityGroup` - Groups claim will contain identifiers of all security groups of which the user is a member.
      - `All` - Groups claim will contain the identifiers of all security groups and distribution lists of which the user is a member.

   d. Click **Manage Manifest**.
   e. Click **Upload Manifest** and select the modified manifest.



Configure a Single Sign-On Service Provider

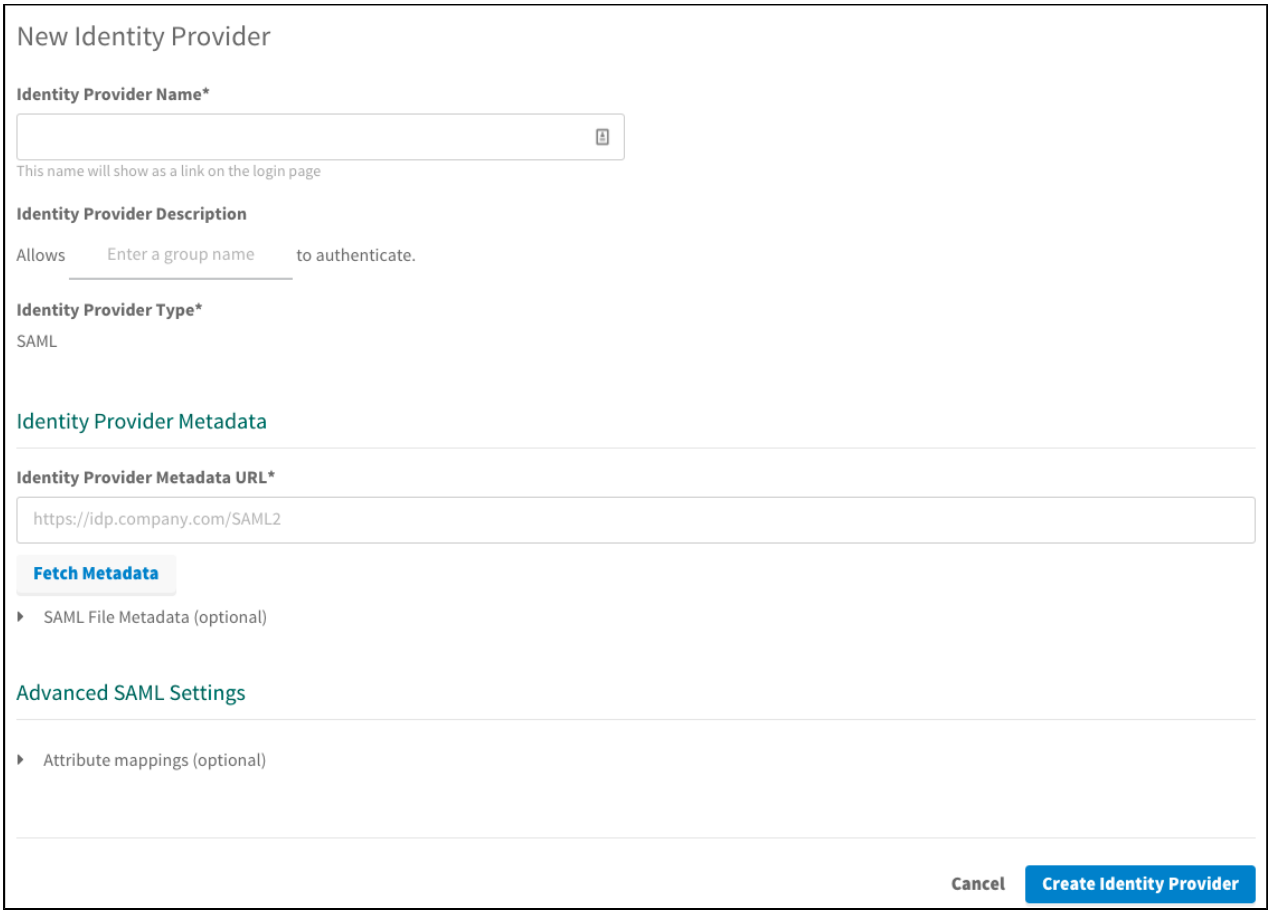
This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Setting up SAML

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **New Identity Provider** to create a new identity provider.



4. To create a new identity provider, perform the following steps:

    a. Enter an identity provider name into **Identity Provider Name**.
    b. (Optional) Enter a description into **Identity Provider Description**.
    c. Click **SAML File Metadata (optional)** follwed by clicking the **Upload Identity Provider Metadata** button to upload your metadata XML.

> **Note**: The Single Sign-On does not support DOS file format imports. Convert the file in one of the following ways:
>
> - Option 1: Execute `dos2unix` on the metadata file.

- Option 2: Create a Unix file, then copy and paste the contents from the downloaded metadata file to the newly created file.

d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

Configure Group Permissions

1. Add groups to be propagated from the external identity provider to the ID token by following these steps:

    a. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
    b. Select your plan and click **Manage Identity Providers** on the drop-down menu.
    c. Click **Group Whitelist** next to your identity provider.
    d. Enter the group names.
    e. Click **Save Group Whitelist**.

2. Map the groups to resources defined in the SSO service by following these steps:

    a. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
    b. Select your plan and click **Manage Identity Providers** on the drop-down menu.
    c. Click **Resource Permissions**.
    d. Click **New Permissions Mapping** and perform the following steps:

        i. Enter a **Group Name**.
        ii. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.
        iii. Click **Save Permissions Mapping**.

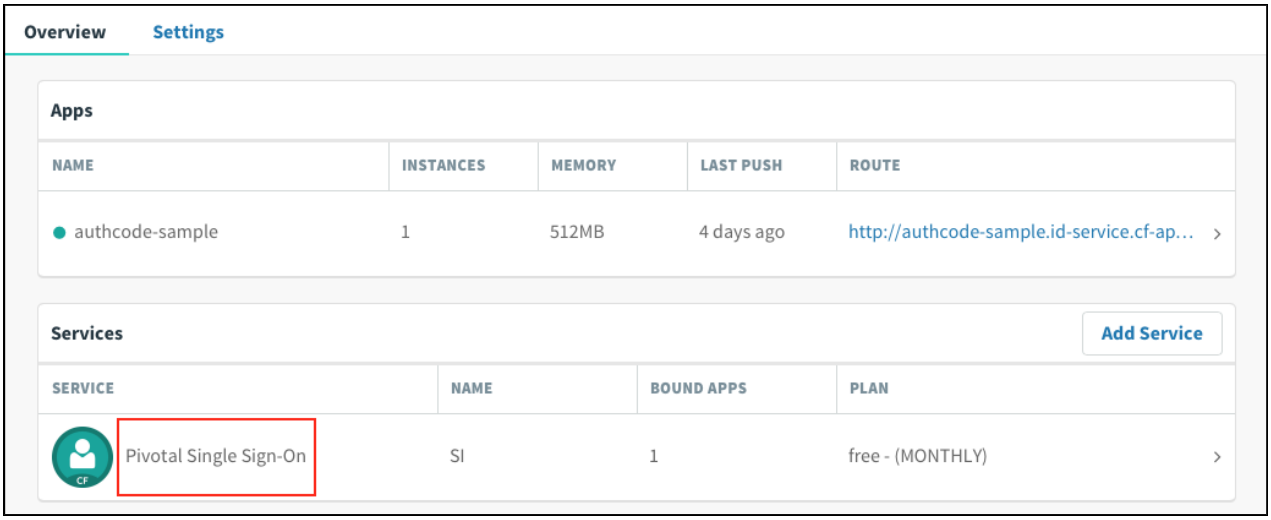
Testing

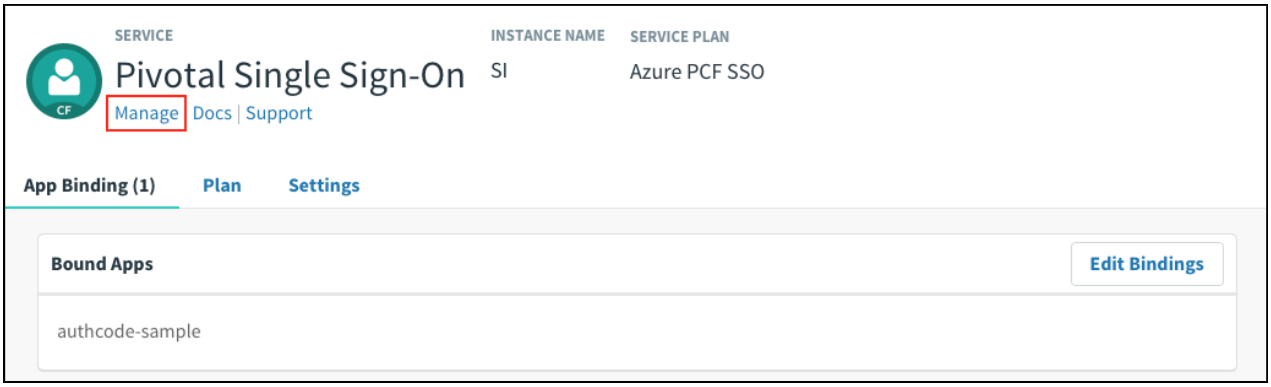
This topic describes how an administrator can test the connection between SSO and Azure Active Directory. An administrator can test both service provider and identity provider connections.

Test Your Service Provider Connection

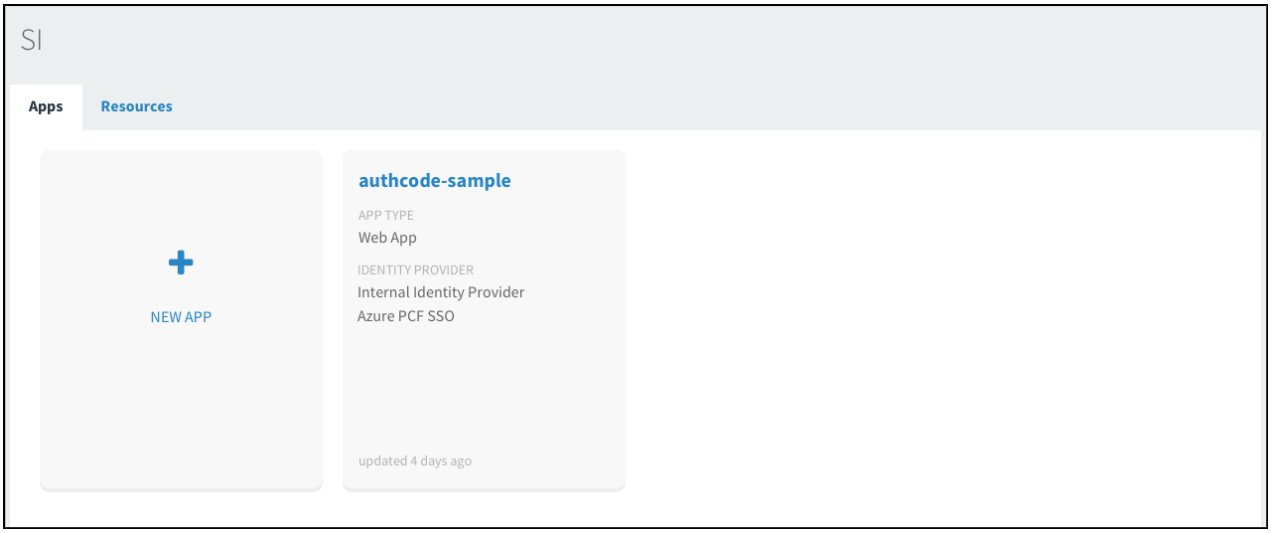
1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN` and navigate to the organization and space where your application is located.

2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Click on the service instance and click **Manage**.

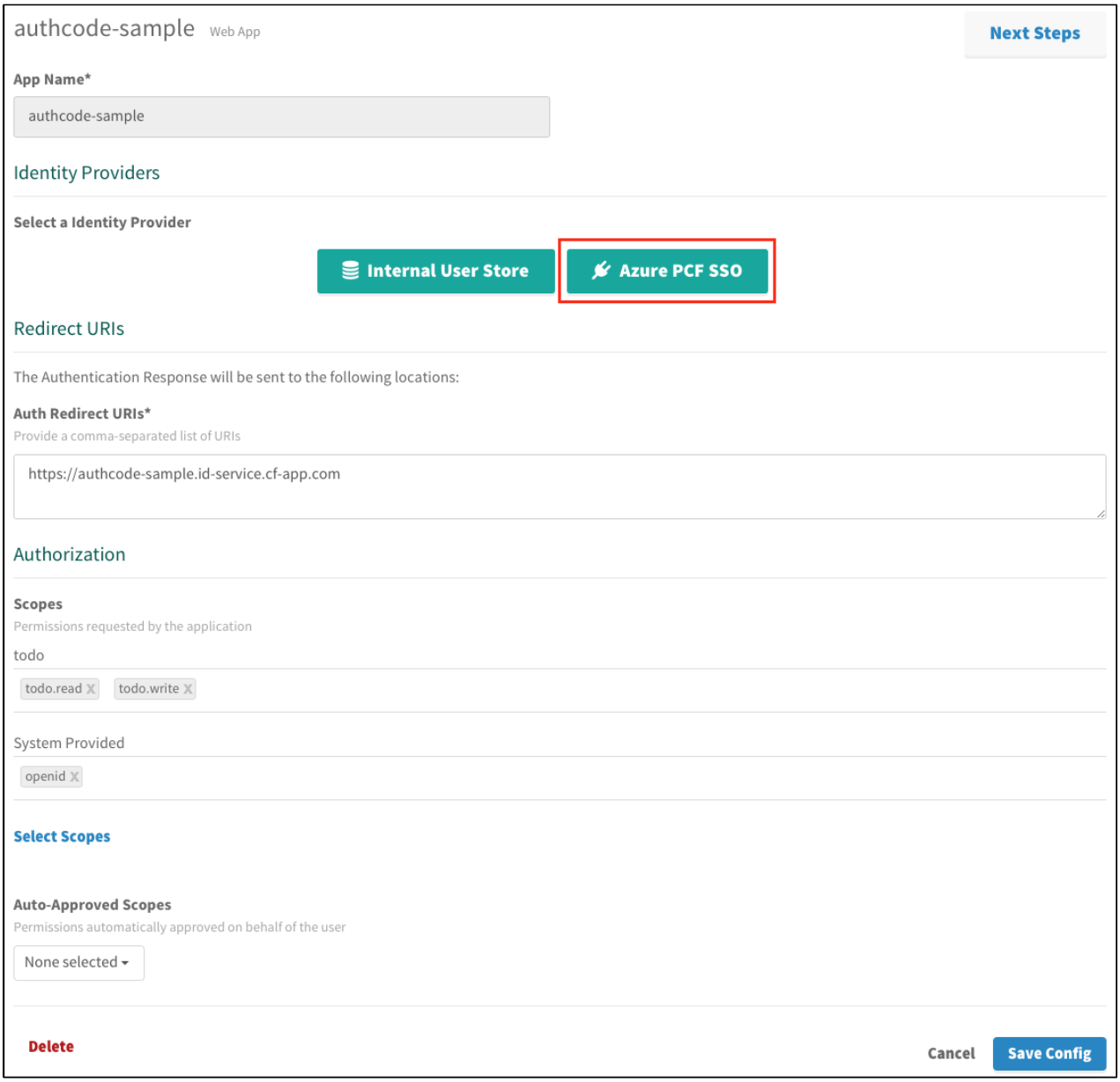




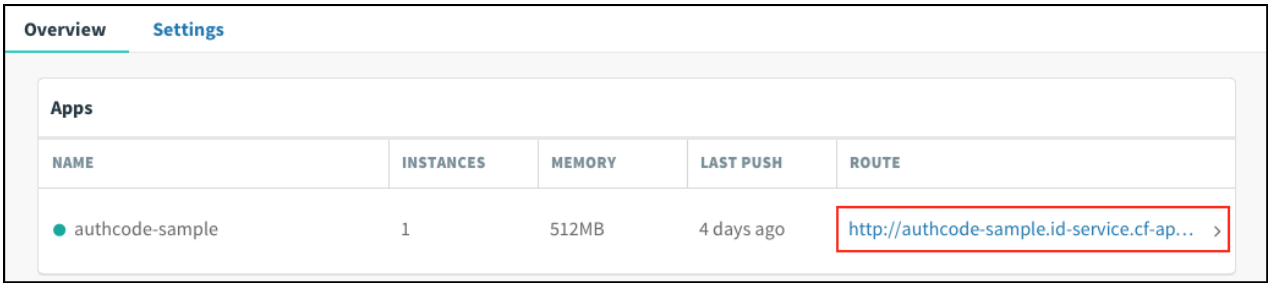
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the Azure AD identity provider.

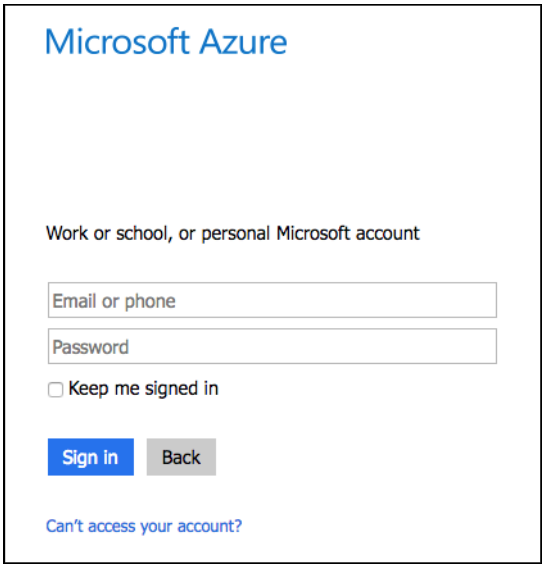


5. Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.

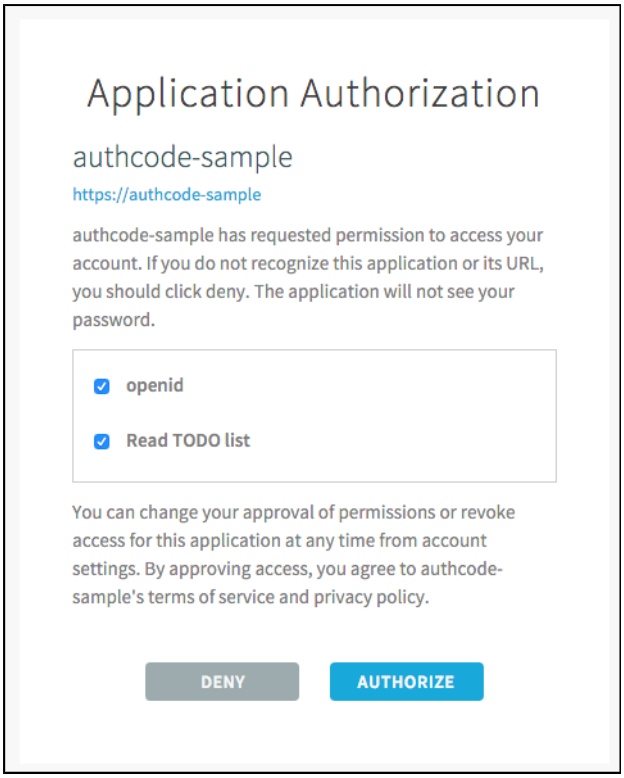


6. Click the link.

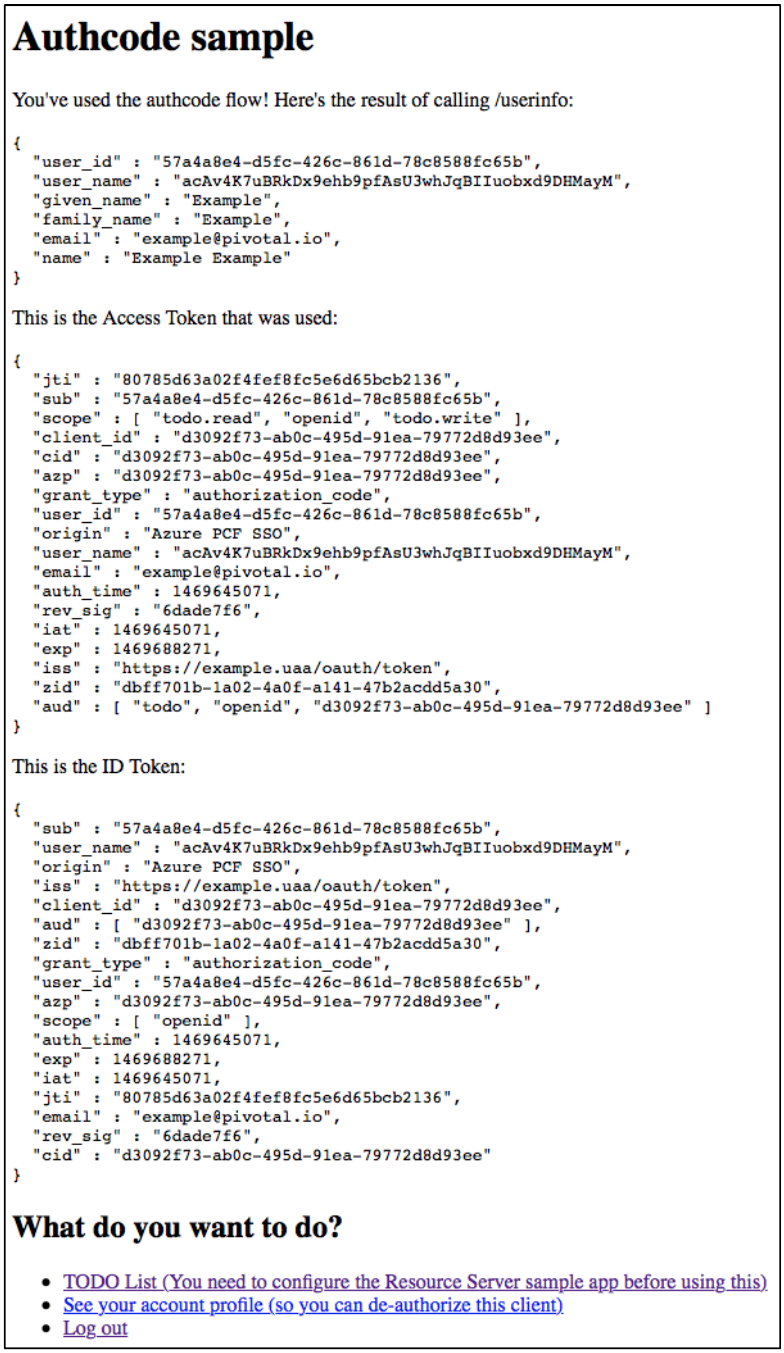
7. On the identity provider sign-in page, enter your credentials and click **Sign In**.



8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. The access token and ID token displays.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "57a4a8e4-d5fc-426c-861d-78c8588fc65b",
  "user_name" : "acAv4K7uBRkDx9ehb9pfAsU3whJqBIIuobxd9DHMayM",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "80785d63a02f4fef8fc5e6d65bcb2136",
  "sub" : "57a4a8e4-d5fc-426c-861d-78c8588fc65b",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "d3092f73-ab0c-495d-91ea-79772d8d93ee",
  "cid" : "d3092f73-ab0c-495d-91ea-79772d8d93ee",
  "azp" : "d3092f73-ab0c-495d-91ea-79772d8d93ee",
  "grant_type" : "authorization_code",
  "user_id" : "57a4a8e4-d5fc-426c-861d-78c8588fc65b",
  "origin" : "Azure PCF SSO",
  "user_name" : "acAv4K7uBRkDx9ehb9pfAsU3whJqBIIuobxd9DHMayM",
  "email" : "example@pivotal.io",
  "auth_time" : 1469645071,
  "rev_sig" : "6dade7f6",
  "iat" : 1469645071,
  "exp" : 1469688271,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "dbff701b-1a02-4a0f-a141-47b2acdd5a30",
  "aud" : [ "todo", "openid", "d3092f73-ab0c-495d-91ea-79772d8d93ee" ]
}
```


This is the ID Token:


```
{
  "sub" : "57a4a8e4-d5fc-426c-861d-78c8588fc65b",
  "user_name" : "acAv4K7uBRkDx9ehb9pfAsU3whJqBIIuobxd9DHMayM",
  "origin" : "Azure PCF SSO",
  "iss" : "https://example.uaa/oauth/token",
  "client_id" : "d3092f73-ab0c-495d-91ea-79772d8d93ee",
  "aud" : [ "d3092f73-ab0c-495d-91ea-79772d8d93ee" ],
  "zid" : "dbff701b-1a02-4a0f-a141-47b2acdd5a30",
  "grant_type" : "authorization_code",
  "user_id" : "57a4a8e4-d5fc-426c-861d-78c8588fc65b",
  "azp" : "d3092f73-ab0c-495d-91ea-79772d8d93ee",
  "scope" : [ "openid" ],
  "auth_time" : 1469645071,
  "exp" : 1469688271,
  "iat" : 1469645071,
  "jti" : "80785d63a02f4fef8fc5e6d65bcb2136",
  "email" : "example@pivotal.io",
  "rev_sig" : "6dade7f6",
  "cid" : "d3092f73-ab0c-495d-91ea-79772d8d93ee"
}
```

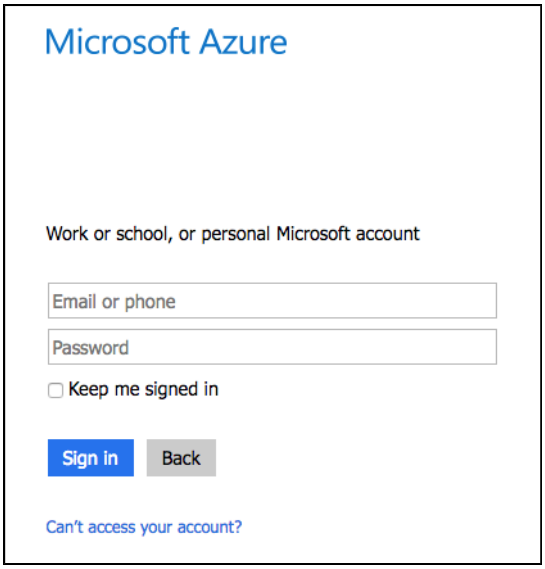

What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

Test Your Identity Provider Connection

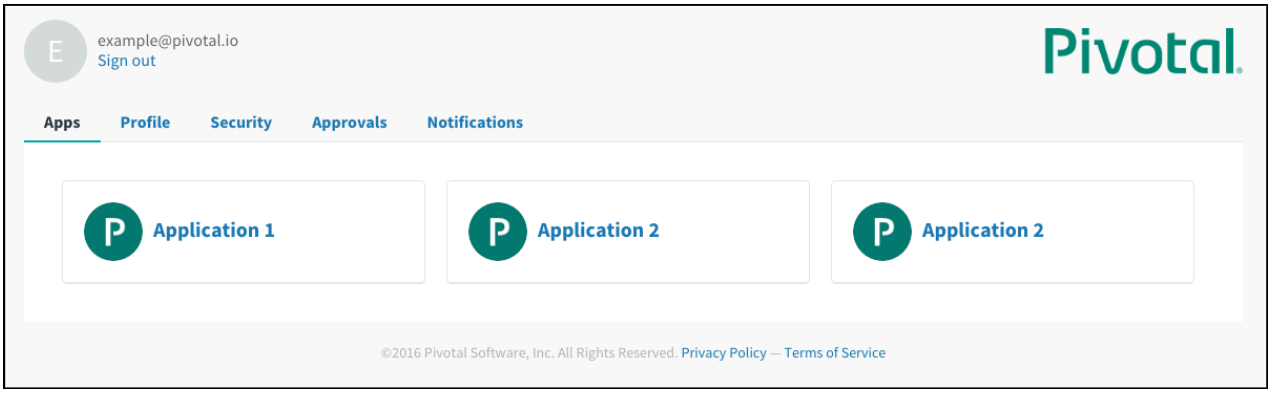
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to Azure AD.



2. Navigate to your application and click it.

3. You are redirected to the page that lists applications you have access to.

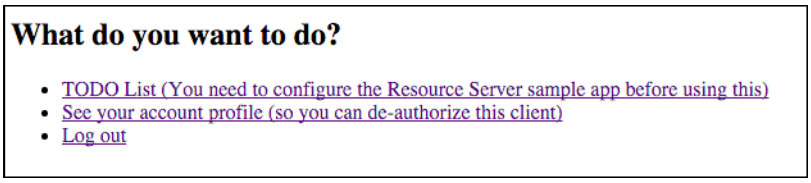


Test Your Single Sign-Off

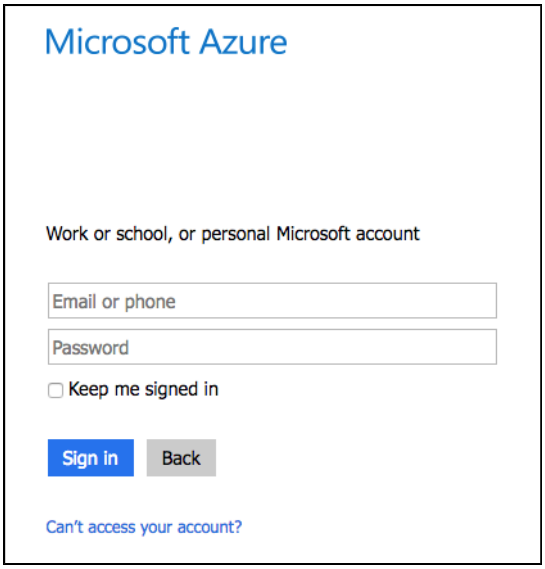
Test single sign-off to ensure that when users log out of the application, they are logged out of Azure AD as well.

1. Sign into the sample application. Information about the access and ID token displays, as well as the "What do you want to do?" section.

2. Under "What do you want to do?", click **Log out**.



3. You are logged out and redirected to the Azure AD login page.



Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between Azure Active Directory and Pivotal Single Sign-On (SSO).

App ID Not Found

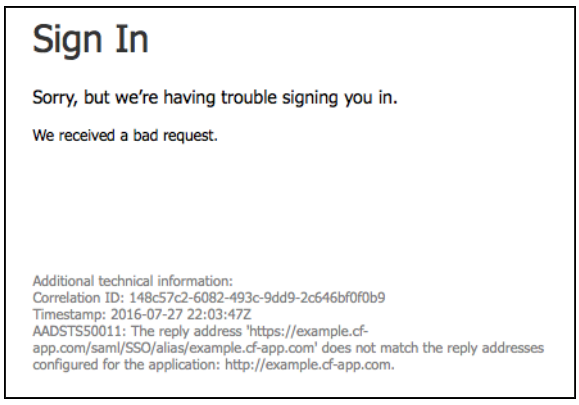
Symptom:

Explanations:

- The App ID URI is misconfigured on Azure AD.

Reply URL Does Not Match

Symptom:



Explanation:

- The Reply URL is misconfigured on Azure AD.

Missing Name ID

Symptom:

Explanation:

- The identity provider metadata has the `RoleDescriptor` elements or is missing configurations for Name ID. See Configure Identity Provider Metadata.

CA Single Sign-On Integration Guide Overview

CA Single Sign-On (formally known as CA SiteMinder) is a Web Access Management system that supports advanced authentication, risk-based security policies, and federated identities. This documentation describes how to configure a single sign-on partnership between CA Single Sign-On as the Identity Provider (IdP) and the Single Sign-On Service (SSO) for Pivotal Cloud Foundry as the Service Provider (SP).

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

Prerequisites

To integrate CA Single Sign-On with Pivotal Cloud Foundry (PCF), you need the following:

Pivotal

- PCF, version 1.7.0 or later
- Single Sign-On, version 1.1.0 or later

CA Single Sign-On

- CA Single Sign-On 12.52
- A Signed Certificate by a Certificate Authority

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic..

CA Single Sign-On Integration Guide

Configuring CA Single Sign-On with SSO

Complete both steps below to integrate your deployment with CA Single Sign-On and SSO.

1. Configure CA Single Sign-On as an Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

- Testing
- Troubleshooting

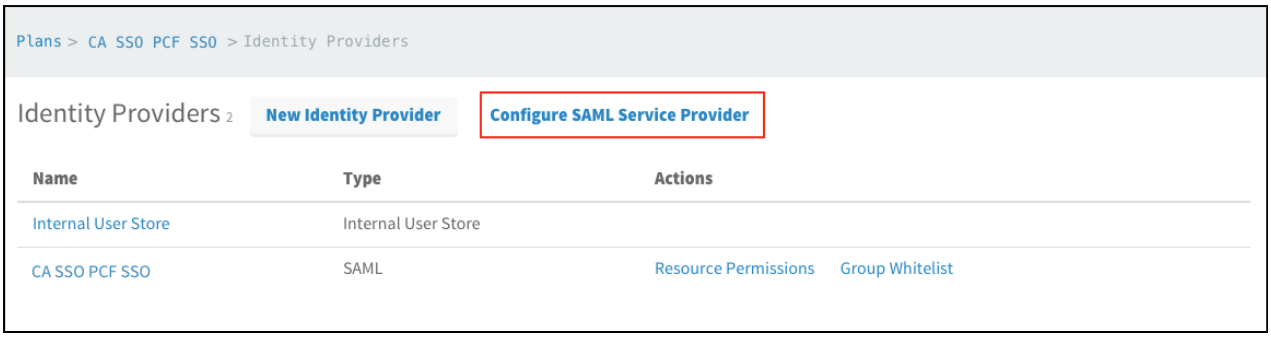
Configure CA Single Sign-On as an Identity Provider

This topic describes how to set up CA Single Sign-On as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and CA Single Sign-On.

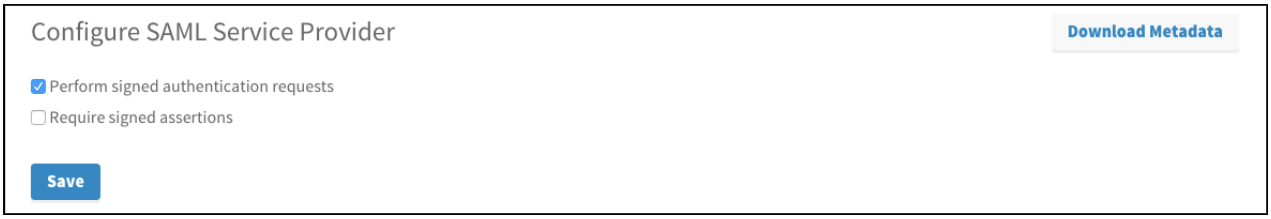
Set up SAML in PCF

1. Log in to the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **Configure SAML Service Provider**.



4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

Set up SAML in CA Single Sign-On

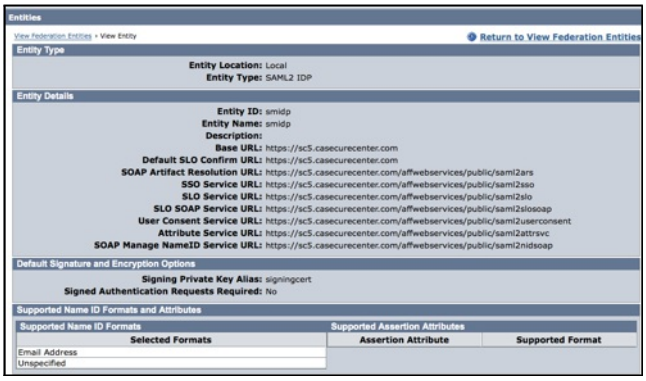
1. Sign in as a CA Single Sign-On administrator.
2. Click the **Federation** tab.
3. Click on the **Entities** link.
4. Click the **Create Entity** button and perform the following steps:
    a. Select **Local** for **Entity Location**.
    b. Select **SAML2 IDP** for **New Entity Type**.

c. Click the **Next** button.

5. In the **Entities** section, perform the following steps:

   a. Enter an **Entity ID**.
   b. Enter an **Entity Name**.
   c. Enter a **Description**.
   d. Enter the fully-qualified domain name for your CA Single Sign-On as the **Base URL**.
   e. Select or import a **Signing Private Key Alias**.
   f. Select a **Name ID format**.
   g. Click the **Next** button.

6. Confirm the Entity Details and click the **Finish** button.



7. Click the **Federation** tab.

8. Click on the **Entities** link.

9. Click the **Import Metadata** button and perform the following steps:

   a. Click **Browse** and select the downloaded metadata for **Metadata file**.
   b. Select **Remote Entity** for **Import As**.
   c. Select **Create New** for **Operation**.
   d. Click the **Next** button.

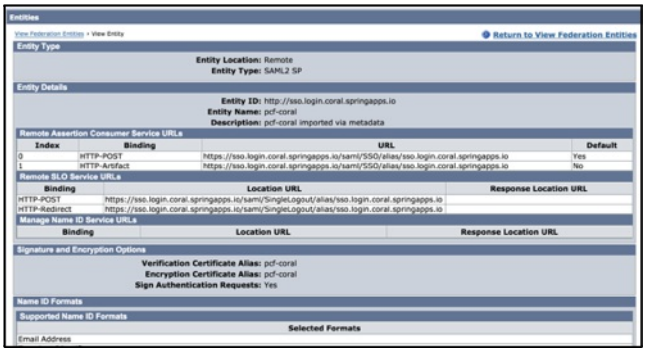
10. In the **Select Entity Defined in Metadata File** section, perform the following steps:

    a. Enter an **Entity Name**.
    b. Click the **Next** button.

11. In the **Select Key Entries to Import** section, perform the following steps:

    a. Enter an **Alias**.
    b. Click the **Next** button.

12. Confirm the Entity Details and click the **Finish** button.

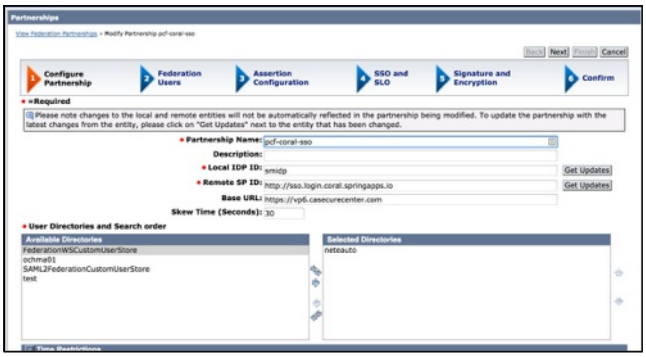


13. Click on the **Federation** tab.

14. Click **Create Partnership** and select **SAML2 IDP -> SP**.

15. In the **Configure Partnership** section, perform the following steps:

a. Enter a **Partnership Name**.
b. Enter a **Description**.
c. Select a previously created local entity for **Local IDP**.
d. Select a previously created remote entity for **Remote SP**.
e. Enter a **Skew Time**.
f. Add any **User Directories**.
g. Click the **Next** button.



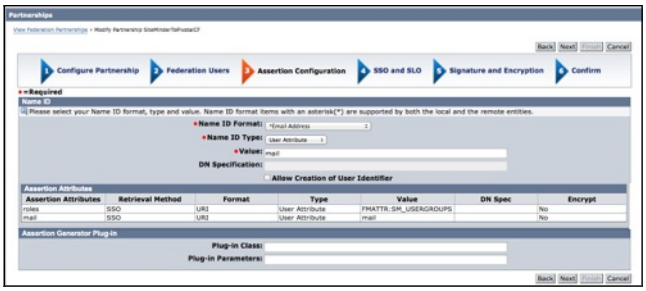
16. Configure **Federation Users** by adding the users you want to include in the partnership and click **Next**.

17. In the **Assertion Configuration** section, perform the following steps:

a. Select a **Name ID Format**.
b. Select **User Attribute** as the **Name ID Type**.
c. Enter `mail` as the **Value**.
d. (Optional) Under **Assertion Attributes**, specify any application or group attributes that you want to map to users in the ID token.

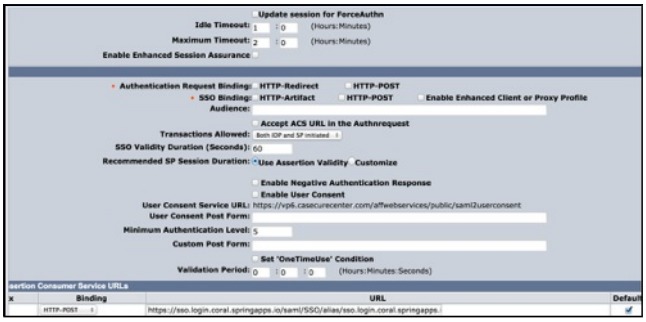
> **Note**: The value for sending a user's groups is **FMATTR:SM_USERGROUPS**.

e. Click the **Next** button.



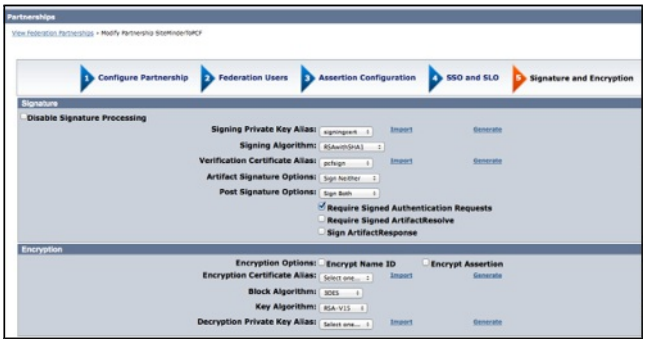
18. In the **SSO and SLO** section, perform the following steps:

a. Enter the **Authentication URL**.
b. Select **HTTP-Post** for **SSO Binding**.
c. Select **Both IDP and SP initiated** for **Transactions Allowed**.
d. Click the **Next** button.



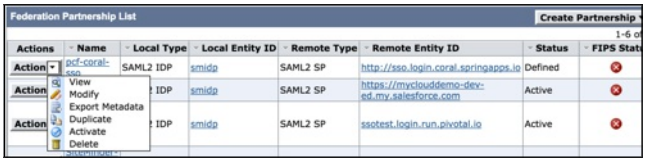
19. In the **Signature and Encryption** section, perform the following steps:

    a. Select your key alias for **Signing Private Key Alias**.
    b. Select your certificate alias for **Verification Certificate Alias**.
    c. Click the **Next** button.



20. Confirm the Partnership Details and click the **Finish** button.

21. Click the **Action** button and click **Activate**.



22. Click the **Action** button and click **Export Metadata**.

Configure a Single Sign-On Service Provider

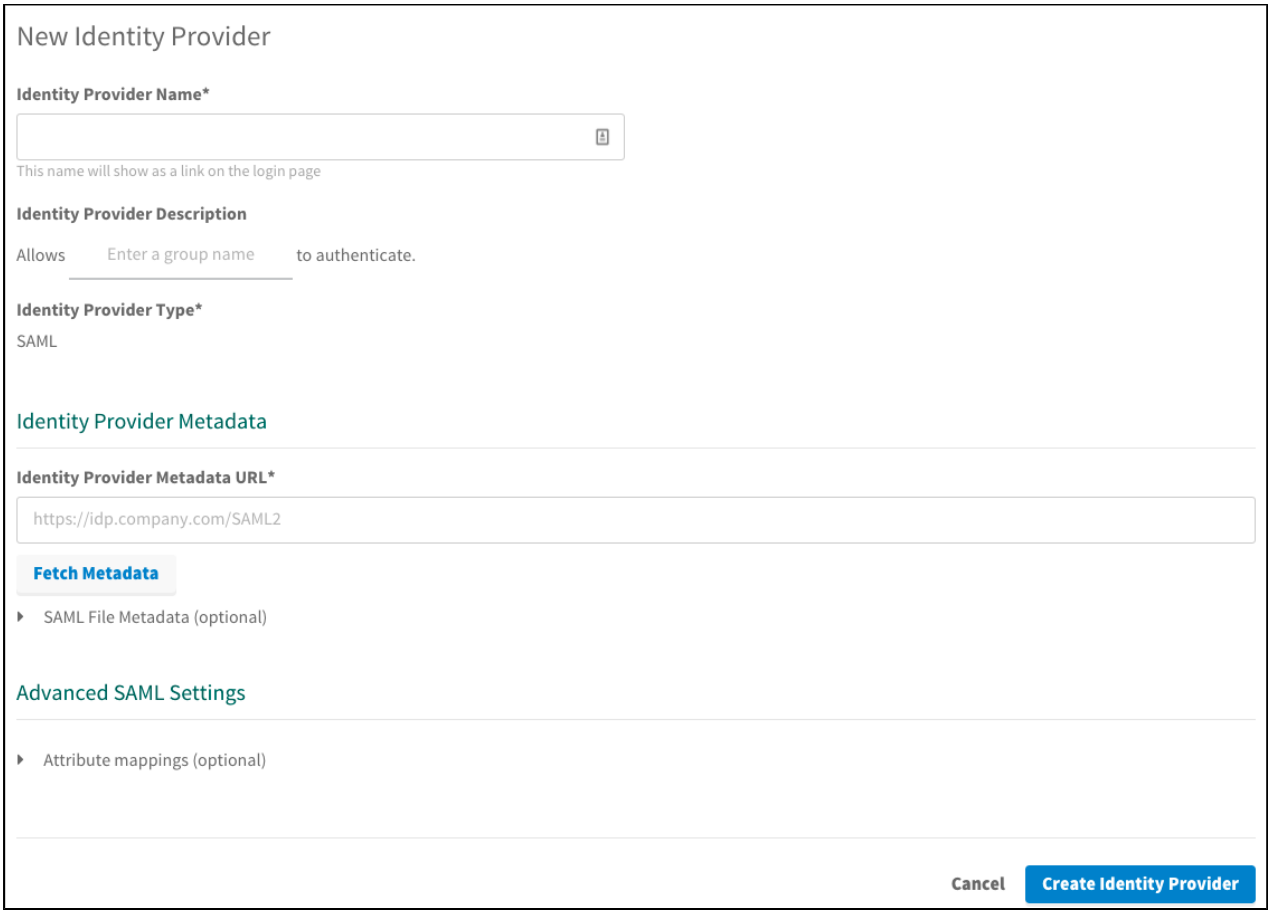
This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Setting up SAML

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **New Identity Provider** to create a new identity provider.



4. To create a new identity provider, perform the following steps:

    a. Enter an identity provider name in **Identity Provider Name**.
    b. (Optional) Enter a description in **Identity Provider Description**.
    c. Click **SAML File Metadata (optional)** followed by clicking the **Upload Identity Provider Metadata** button to upload your metadata XML.
    d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

6. Click **Resource Permissions**.

7. Click **New Permissions Mapping** and perform the following steps:

    a. Enter a **Group Name**.
    b. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.

8. Navigate to the identity provider list.

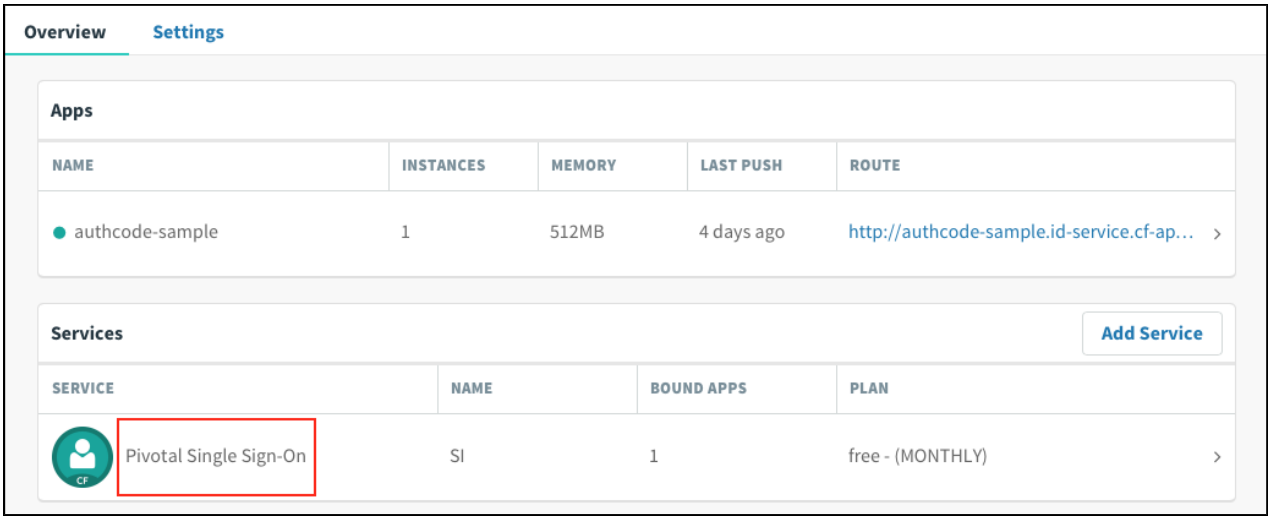
9. Click **Group Whitelist** and enter the group names from the external identity provider that should be propagated in the ID token.

Testing

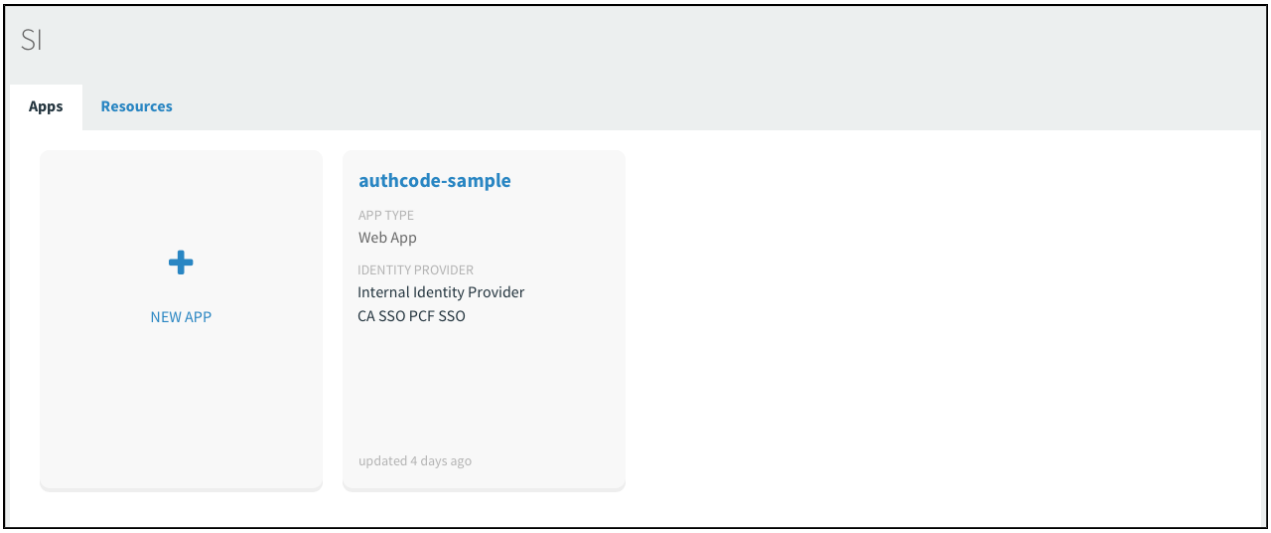
This topic describes how an administrator can test the connection between SSO and CA Single Sign-On. An administrator can test both service provider and identity provider connections.

Test Your Service Provider Connection

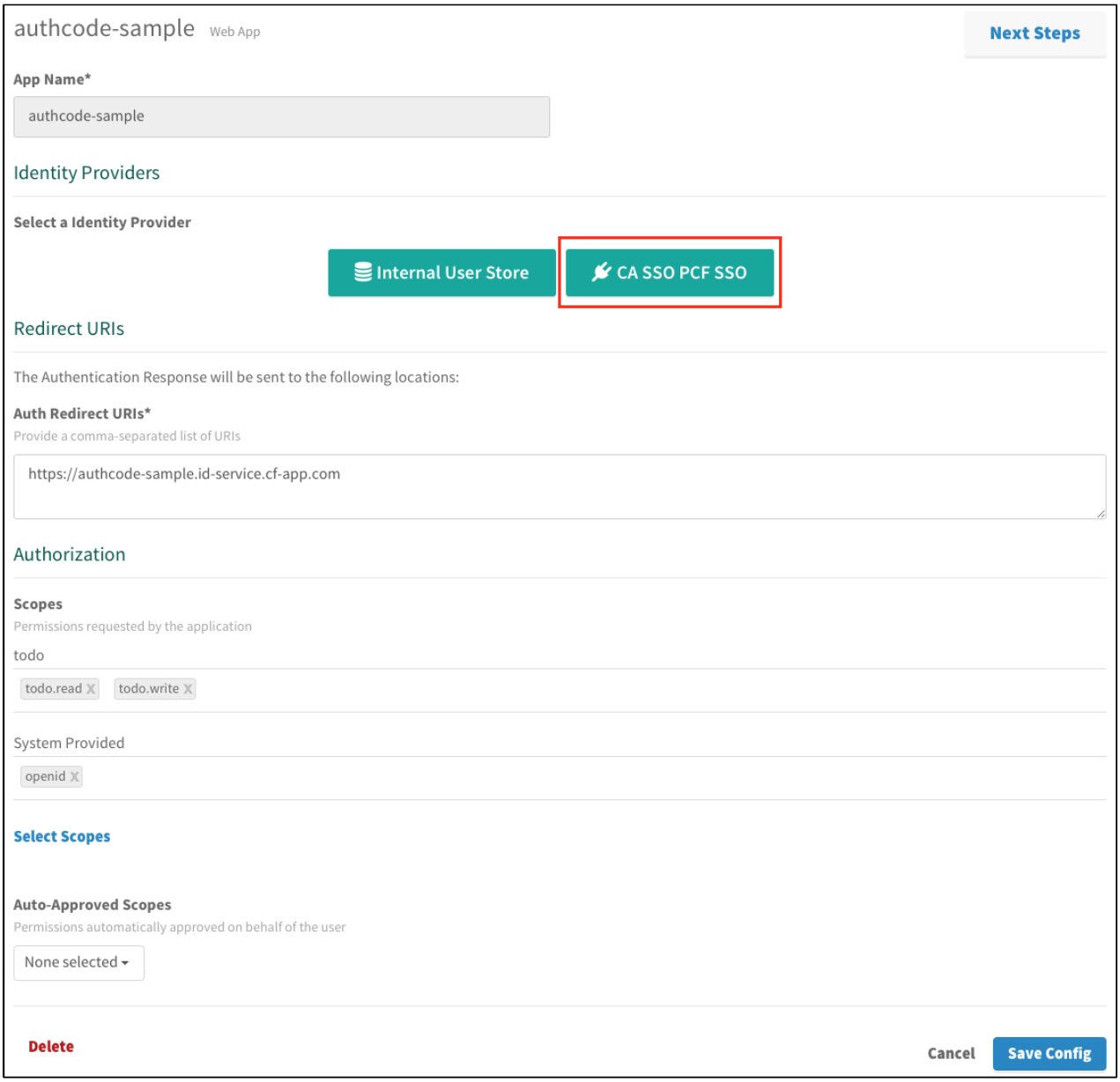
1. Log in to Apps Manager at https://apps.YOUR-SYSTEM-DOMAIN and navigate to the organization and space where your application is located.
2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Select the service instance and click **Manage**.



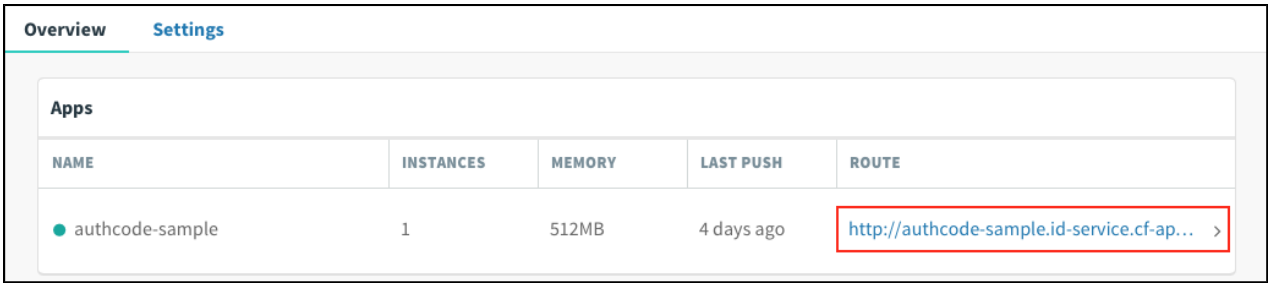
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the CA Single Sign-On identity provider.

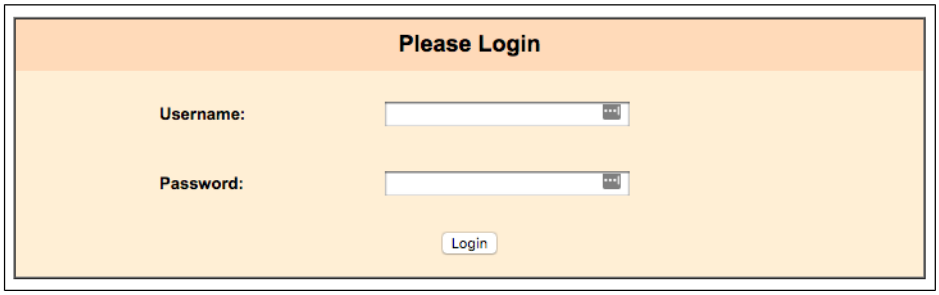


5. Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.

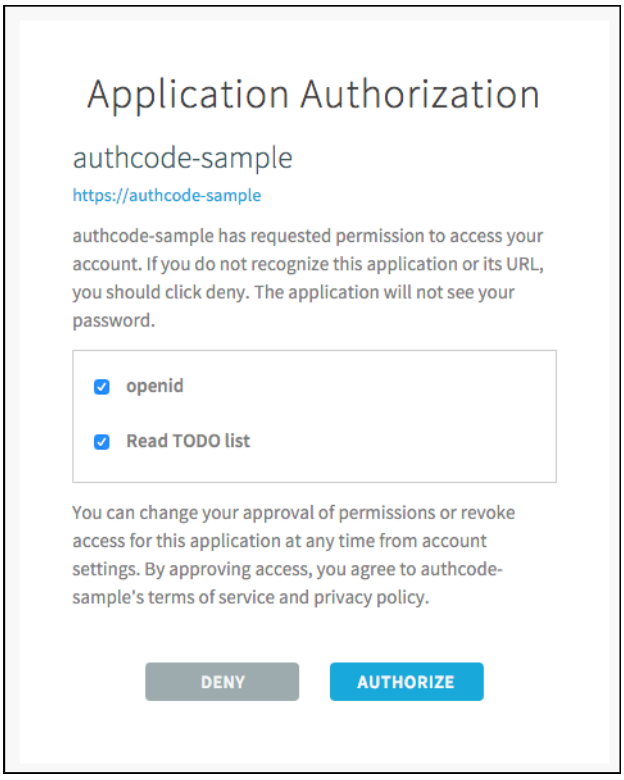


6. Click the link.

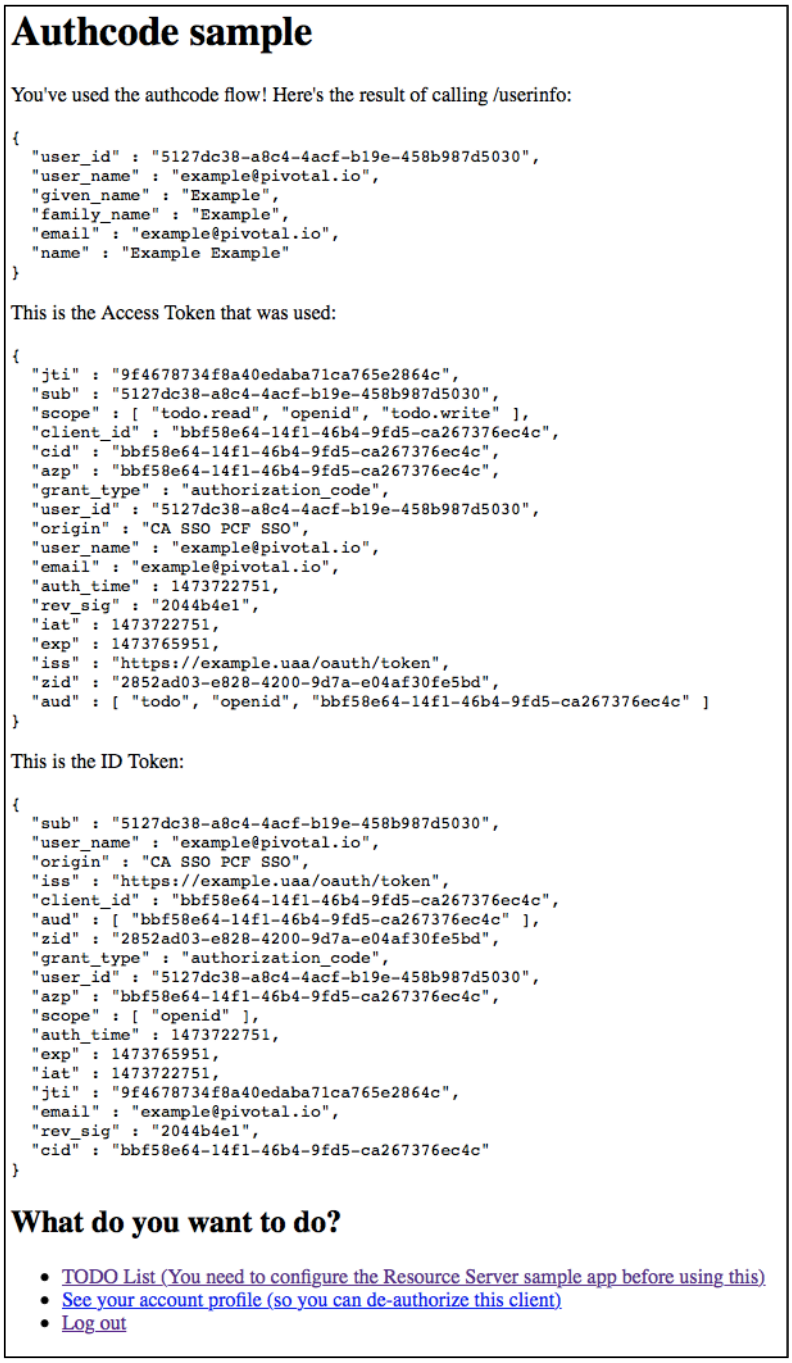
7. On the identity provider sign-in page, enter your credentials and click **Sign On**.



8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. The access token and ID token displays.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "5127dc38-a8c4-4acf-b19e-458b987d5030",
  "user_name" : "example@pivotal.io",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "9f4678734f8a40edaba71ca765e2864c",
  "sub" : "5127dc38-a8c4-4acf-b19e-458b987d5030",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "cid" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "azp" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "grant_type" : "authorization_code",
  "user_id" : "5127dc38-a8c4-4acf-b19e-458b987d5030",
  "origin" : "CA SSO PCF SSO",
  "user_name" : "example@pivotal.io",
  "email" : "example@pivotal.io",
  "auth_time" : 1473722751,
  "rev_sig" : "2044b4e1",
  "iat" : 1473722751,
  "exp" : 1473765951,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "2852ad03-e828-4200-9d7a-e04af30fe5bd",
  "aud" : [ "todo", "openid", "bbf58e64-14f1-46b4-9fd5-ca267376ec4c" ]
}
```


This is the ID Token:


```
{
  "sub" : "5127dc38-a8c4-4acf-b19e-458b987d5030",
  "user_name" : "example@pivotal.io",
  "origin" : "CA SSO PCF SSO",
  "iss" : "https://example.uaa/oauth/token",
  "client_id" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "aud" : [ "bbf58e64-14f1-46b4-9fd5-ca267376ec4c" ],
  "zid" : "2852ad03-e828-4200-9d7a-e04af30fe5bd",
  "grant_type" : "authorization_code",
  "user_id" : "5127dc38-a8c4-4acf-b19e-458b987d5030",
  "azp" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c",
  "scope" : [ "openid" ],
  "auth_time" : 1473722751,
  "exp" : 1473765951,
  "iat" : 1473722751,
  "jti" : "9f4678734f8a40edaba71ca765e2864c",
  "email" : "example@pivotal.io",
  "rev_sig" : "2044b4e1",
  "cid" : "bbf58e64-14f1-46b4-9fd5-ca267376ec4c"
}
```

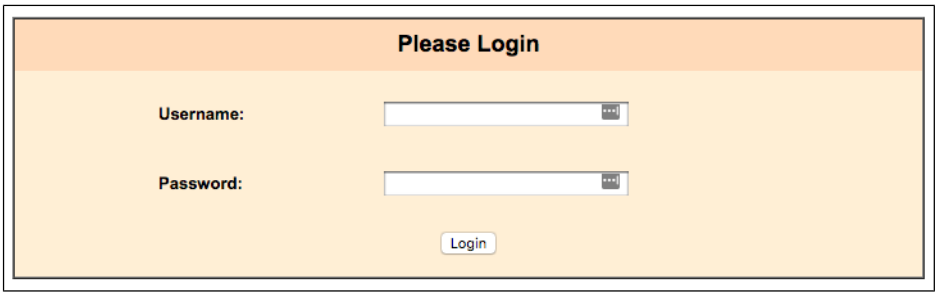

What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

Test Your Identity Provider Connection

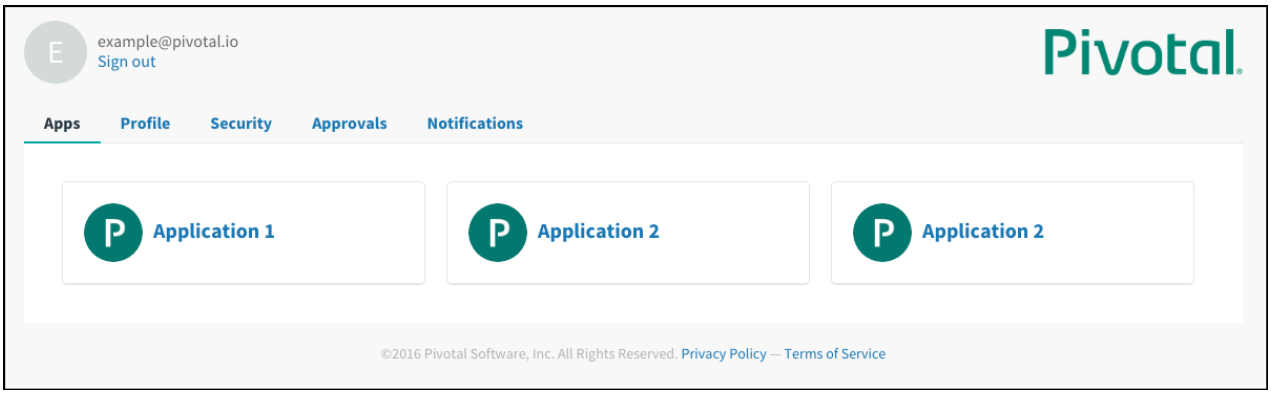
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to CA Single Sign-On.



2. Navigate to your application and click it.

3. You are redirected to the page that lists applications you have access to.

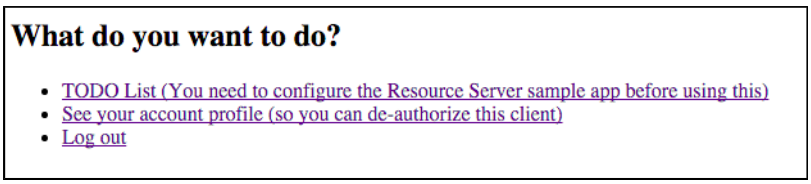


Test Your Single Sign-Off

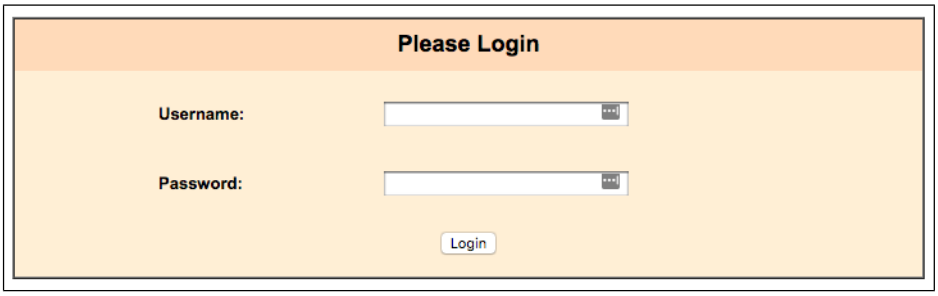
Test single sign-off to ensure that when users log out of the application, they are logged out of CA Single Sign-On as well.

1. Sign in to the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

2. Under “What do you want to do?”, click **Log out**.



3. You are logged out and redirected to the CA Single Sign-On login page.

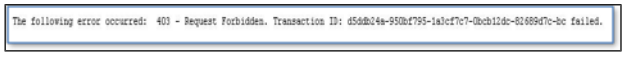


Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between PingOne Cloud and Pivotal Single Sign-On (SSO).

CA Single Sign-On Partnership is Inactive

Symptom:

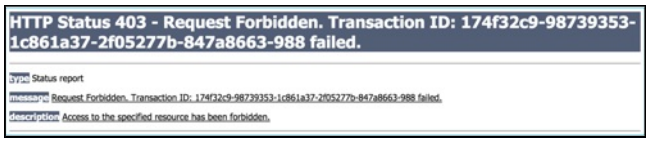


Explanations:

- The CA Single Sign-On is inactive in CA Single Sign-On.

Service Provider Entity ID Misconfigured

Symptom:

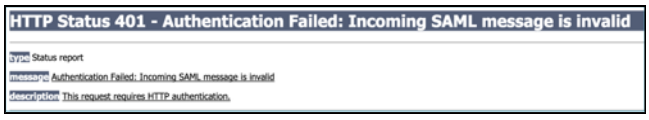


Explanation:

- The service provider Entity ID is misconfigured in CA Single Sign-On.

Incoming SAML message is invalid

Symptom:



Explanation:

- The identity provider Entity ID is misconfigured in CA Single Sign-On or in PCF Single Sign-On.
- The Name ID Format was misconfigured in CA Single Sign-On

Assertion Consumer Service URL Misconfigured

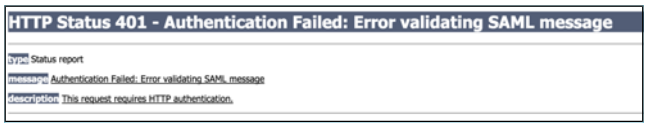
Symptom:

Explanation:

- The service provider Assertion Consumer Service (ACS) is misconfigured in CA Single Sign-On.

Audience Field Misconfigured

Symptom:

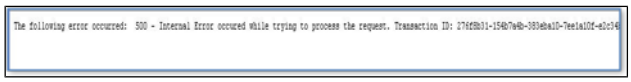


Explanation:

- The service provider Audience Field is misconfigured in CA Single Sign-On.

Expired Certificate

Symptom:



Explanation:

- The certificate has expired in CA Single Sign-On.

Identity Provider SSO URL Misconfigured

Symptom:

Explanation:

- The identity provider SSO URL is misconfigured in PCF Single Sign-On.

Google Cloud Platform OIDC Integration Guide Overview

This documentation describes how to set up the Pivotal Cloud Foundry (PCF) Single Sign-On service to use Google Cloud Platform (GCP) as an OpenID Connect (OIDC) identity provider.

GCP lets you build and host applications and websites, store data, and analyze data on Google's scalable infrastructure.

Prerequisites

To integrate Google Cloud Platform as a single sign-on identity provider for PCF apps, you need:

- **Pivotal**
  - PCF v1.11.0 or later
  - SSO v1.4.1 or later installed on your PCF deployment
  - An SSO service plan configured with plan administrators who manage it and orgs to use it. For help configuring plans, see Manage Service Plans.
- **Google Cloud Platform**
  - An active Google Cloud project
  - A GCP user account with project editor or higher privileges

Integrate Google Cloud Platform OIDC for SSO

Complete the step below to set up GCP as an OIDC identity provider for the SSO service.

1. Configure GCP as an OIDC Identity Provider

Test and Troubleshoot

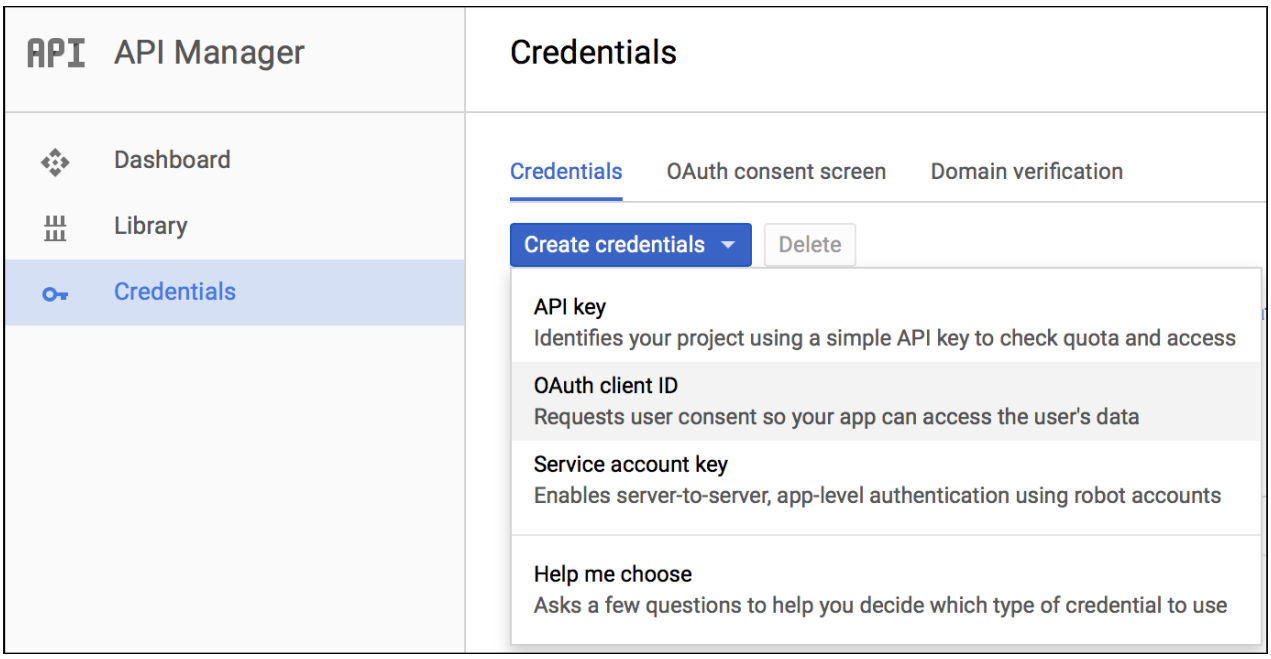
- Testing
- Troubleshooting

Configure GCP as an OIDC Identity Provider

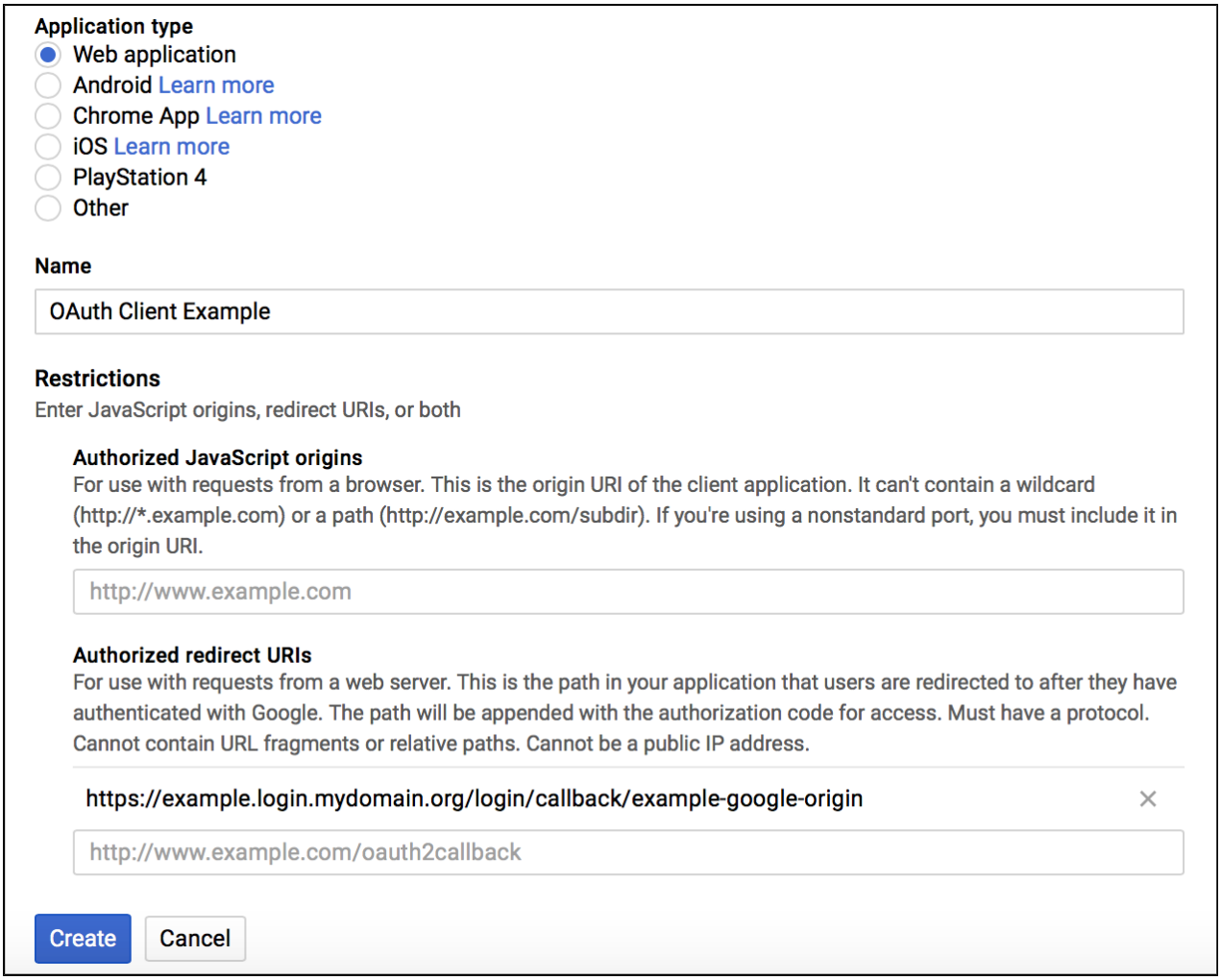
This topic describes how to set up Google Cloud Platform (GCP) as an identity provider for a Single Sign-On (SSO) service plan by configuring OpenID Connect (OIDC) integration in both Pivotal Cloud Foundry (PCF) and GCP.

Generate GCP Client Credentials

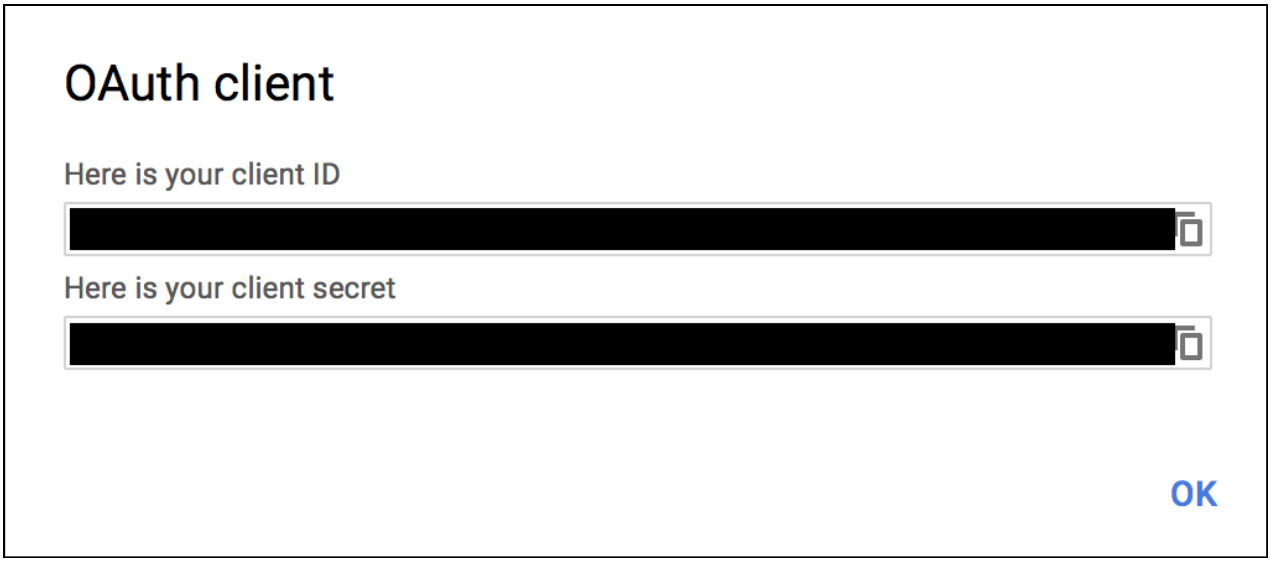
1. Log in to your Google Cloud Platform console.
2. Under the **Credentials** tab, click **Create credentials** > **OAuth client ID**.



3. In the configuration pane that appears, select **Web application** under **Application type** and enter any **Name**. Under **Restrictions**, leave **Authorized JavaScript Origins** blank and for **Authorized redirect URIs** enter a redirect URI of the form `https://AUTH_DOMAIN/login/callback/ORIGIN_KEY`, where:
    - `AUTH_DOMAIN` is the full URL generated based on the **Auth Domain** setting you entered when you created the service plan that you are integrating with GCP.
    - `ORIGIN_KEY` is based on the **Identity Provider Name** you set in the SSO dashboard in Set Up OIDC Identity Provider in SSO below. This value should have no spaces or uppercase letters. You might need to change this value later.



4. Click **Create** and record the **client ID** and **client secret** generated. You will enter these values as your **Relying Party OAuth Client ID** and **Relying Party OAuth Client Secret** in the SSO dashboard in Set Up OIDC Identity Provider in SSO below.



Set Up OIDC Identity Provider in SSO

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` using your UAA administrator credentials. You can find these credentials in your Elastic Runtime tile in Ops Manager under the **Credentials** tab.

2. Click the plan name and select **Manage Identity Providers** from the drop-down menu.

3. Click **New Identity Provider**.

4. Enter an **Identity Provider Name**. This value in all lowercase with dashes replacing spaces becomes your **Origin Key**. For example, `Example Google Origin` becomes `example-google-origin`. If you did not enter this for your OAuth Client's authorized redirect URIs, go back and edit the value in Google Cloud Platform.

5. Enter a **Description**. Space developers see this description when they select an identity provider for their app.

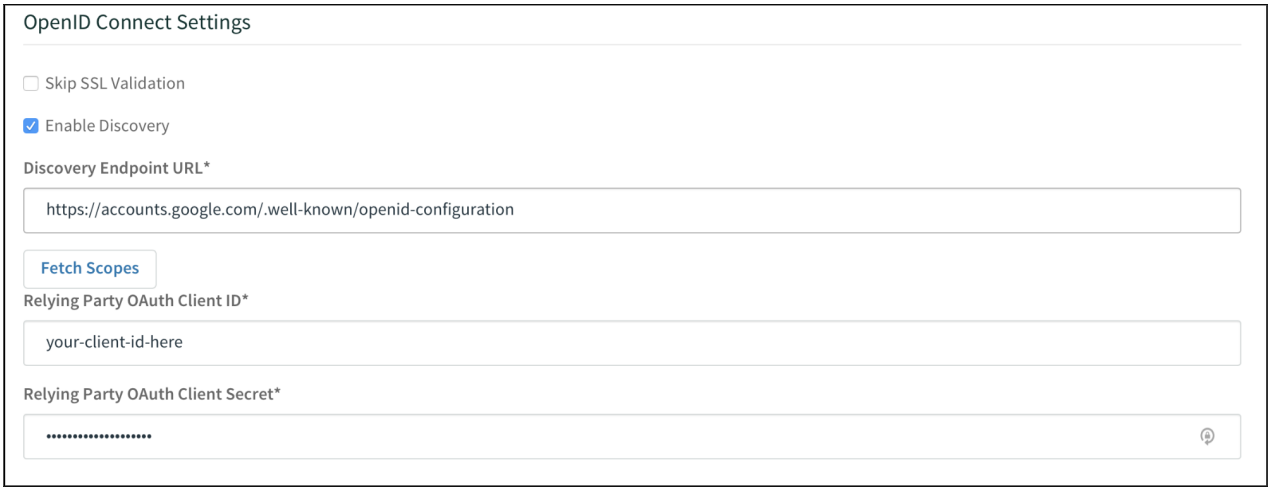
6. Select **OpenID Connect** as the **Identity Provider type**.

7. Make sure the **Enable Discovery** checkbox is selected, to enable OIDC discovery.

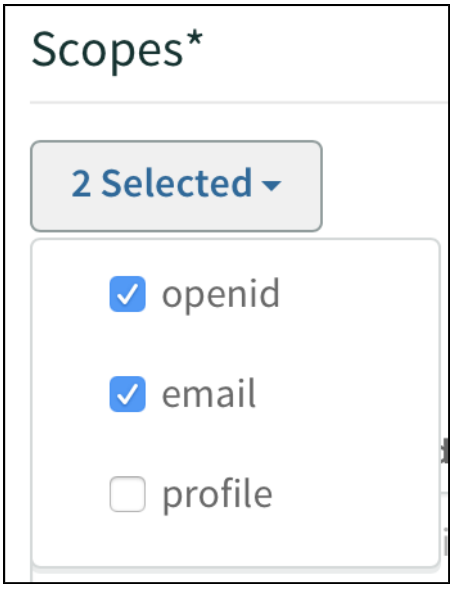
8. For **Discovery Endpoint URL**, enter `https://accounts.google.com/.well-known/openid-configuration`.

9. Click **Fetch Scopes**.

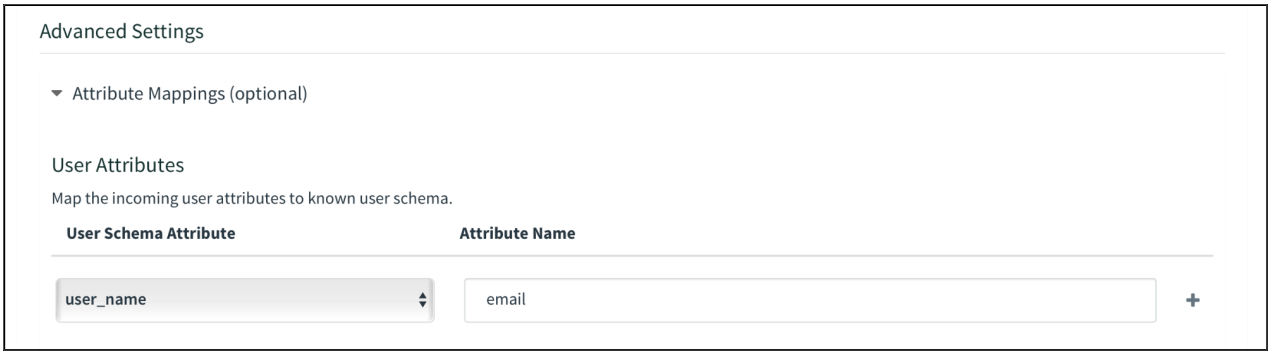
10. Enter your **Relying Party OAuth Client ID** and **Relying Party OAuth Client Secret** from the Generate GCP Client Credentials above.



11. Make sure that `openid` and `email` are selected as scopes. You can select additional scopes if you want.



12. Under **Advanced Settings** > **User Attributes**, map `user_name` to `email`. This enables SSO to identify the authenticated user.



13. (Optional) Configure additional attribute mappings.

14. Click **Create Identity Provider** to save your settings.

15. (Optional) Enable identity provider discovery for the service plan.

Testing

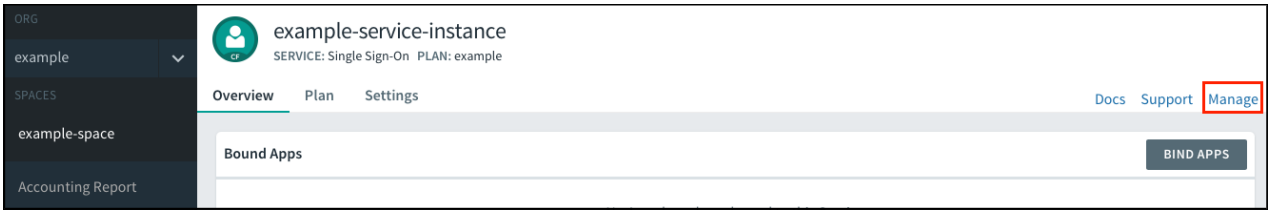
This topic describes how a Pivotal Cloud Foundry (PCF) administrator can test the OpenID Connect (OIDC) connection between the Single Sign-On (SSO) service and Google Cloud Platform.

Test Your Single Sign-On Connection

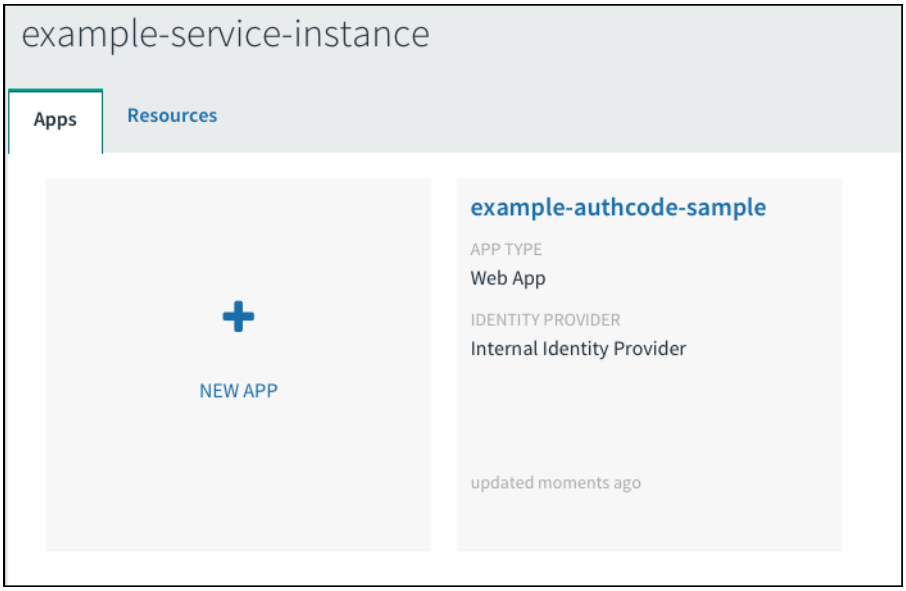
1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN` and navigate to the org and space where your app is located.

2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your app.

3. Select the service instance and click **Manage**.

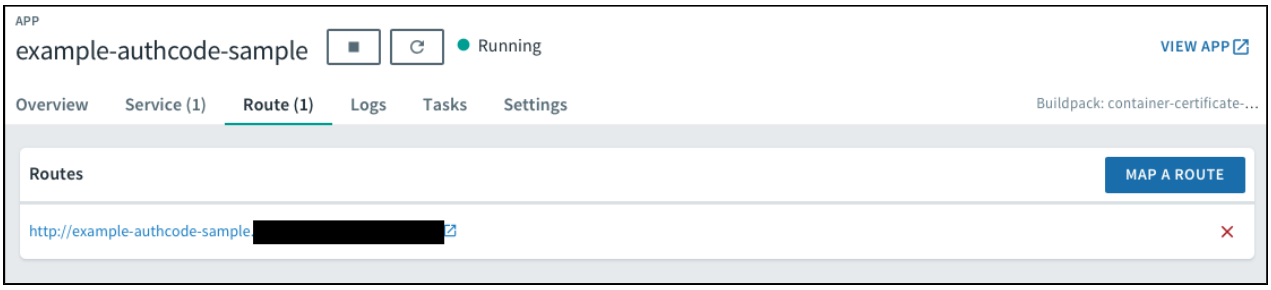


4. Under the **Apps** tab, select your app.



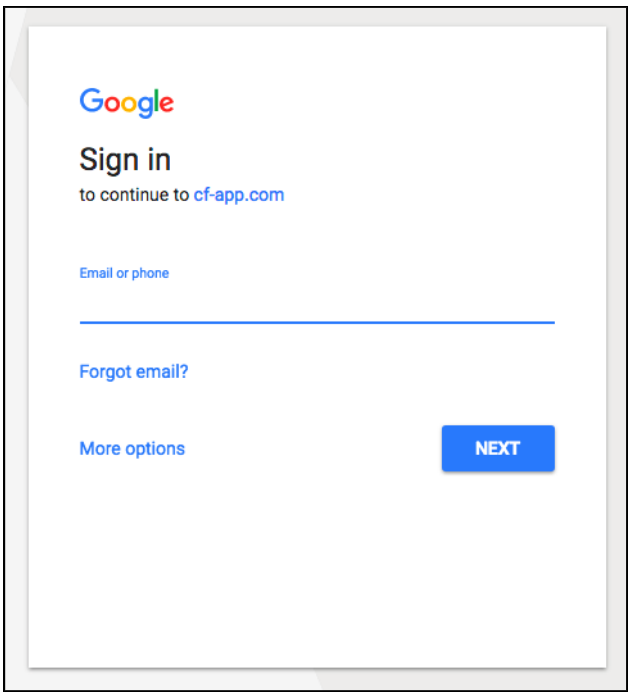
5. Under **Identity Providers**, select the GCP identity provider. Remove any other identity providers.

6. Return to Apps Manager and click the URL listed below your app to access your application.



7. Navigate to your login. You will be redirected to the identity provider to authenticate.

8. On the identity provider sign-in page, enter your credentials and sign in.



9. If the app prompts for authorization to the necessary scopes, click **Authorize**.

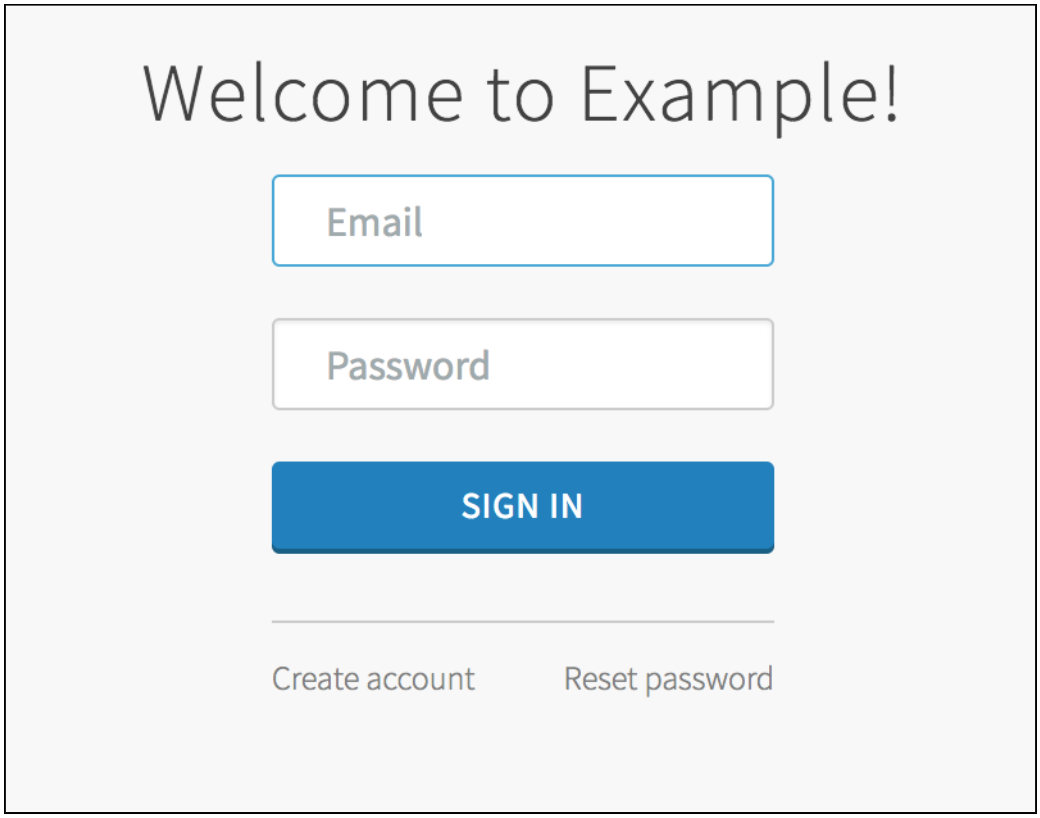
If you are now logged in to your app, your GCP OIDC to SSO connection works.

Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between Google Cloud Platform (GCP) OpenID Connect (OIDC) and Pivotal Single Sign-On (SSO).

No Link for OIDC

Symptom:

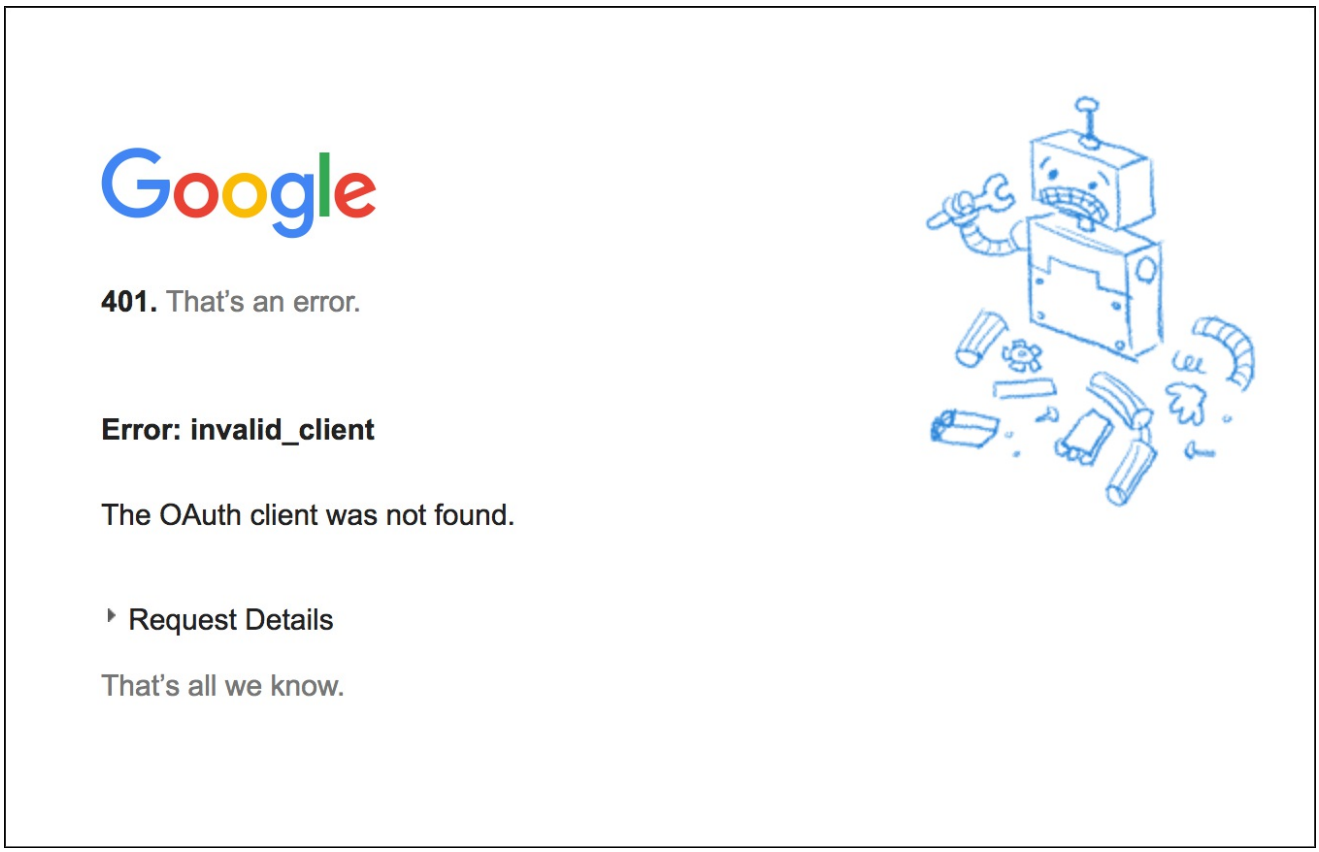


Explanation:

- Incorrect or unavailable discovery URL. No link will appear on the login page.

No OAuth Client Found

Symptom:

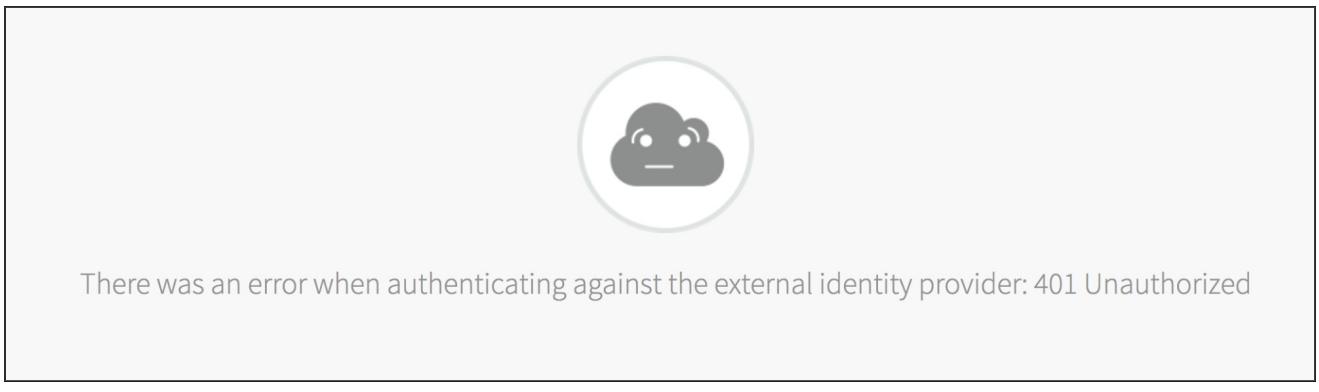


Explanation:

- Incorrect OAuth Client ID configured.

Unauthorized

Symptom:

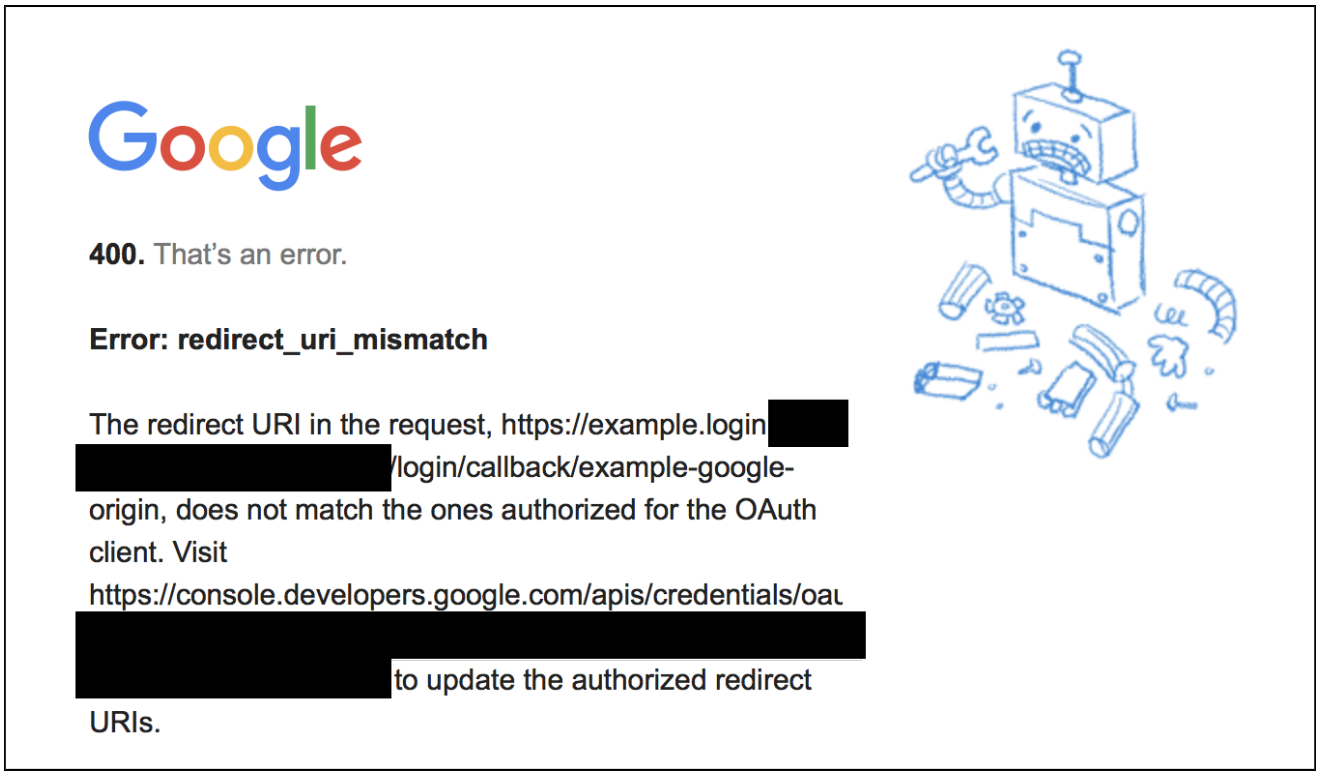


Explanation:

- Incorrect OAuth client secret configured.

Redirect URI Mismatch

Symptom:

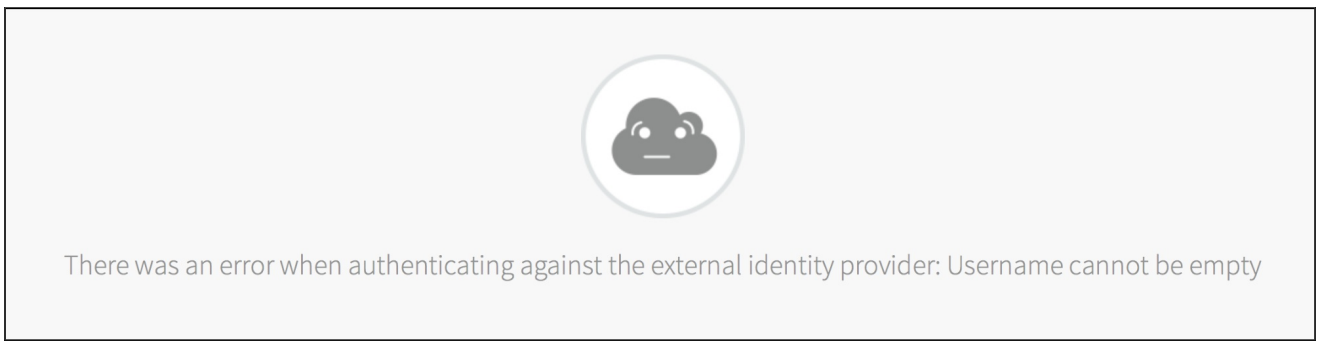


Explanation:

- Incorrect authorization redirect URI on OAuth Client.

Empty Username

Symptom:

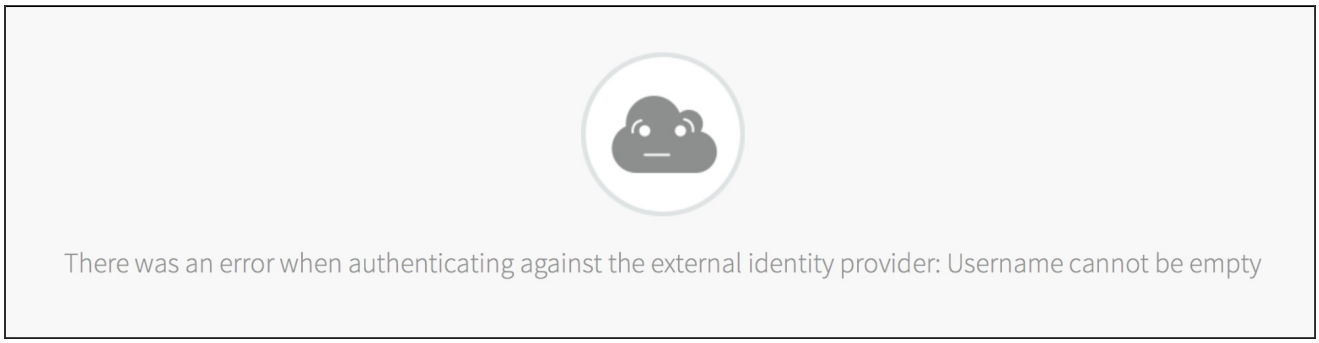


Explanation:

- `user_name` attribute was not mapped to `email`.

Unable to map claim to a username

Symptom:



Explanation:

- The scope for “email” was not configured. Select the “email” scope in your identity provider configurations.

Okta Integration Guide Overview

Okta is an enterprise identity management and single sign-on service that integrates with applications in the cloud, on-premises, or on a mobile device. This documentation describes how to configure a single sign-on partnership between Okta as the Identity Provider (IdP) and the Single Sign-On Service (SSO) for Pivotal Cloud Foundry as the Service Provider (SP).

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

Prerequisites

To integrate Okta with Pivotal Cloud Foundry (PCF), you need:

Pivotal

- PCF, version 1.7.0 or later.
- Single Sign-On, version 1.1.0 or later.

Okta

- Okta, version 2016.07 or later.
- A user with Application Admin privileges.

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic.

Okta Integration Guide

Configuring Okta with SSO

Complete both steps below to integrate your deployment with Okta and SSO.

1. Configure Okta as an Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

- Testing
- Troubleshooting

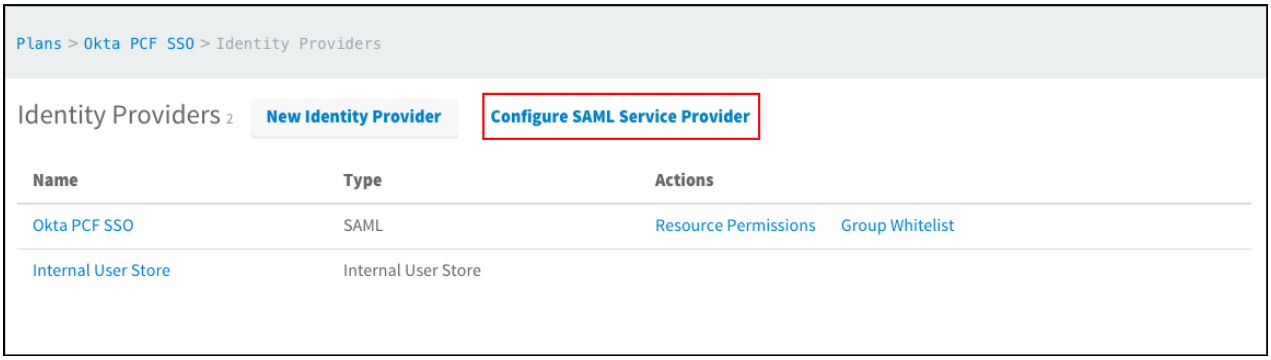
Configure Okta as an Identity Provider

This topic describes how to set up Okta as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and Okta.

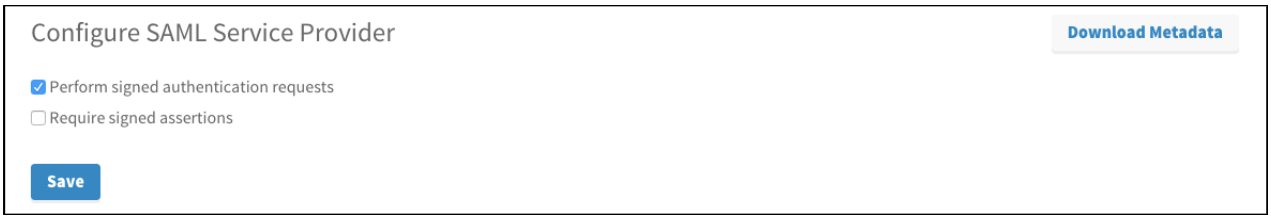
Set up SAML in PCF

1. Log in to the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **Configure SAML Service Provider**.



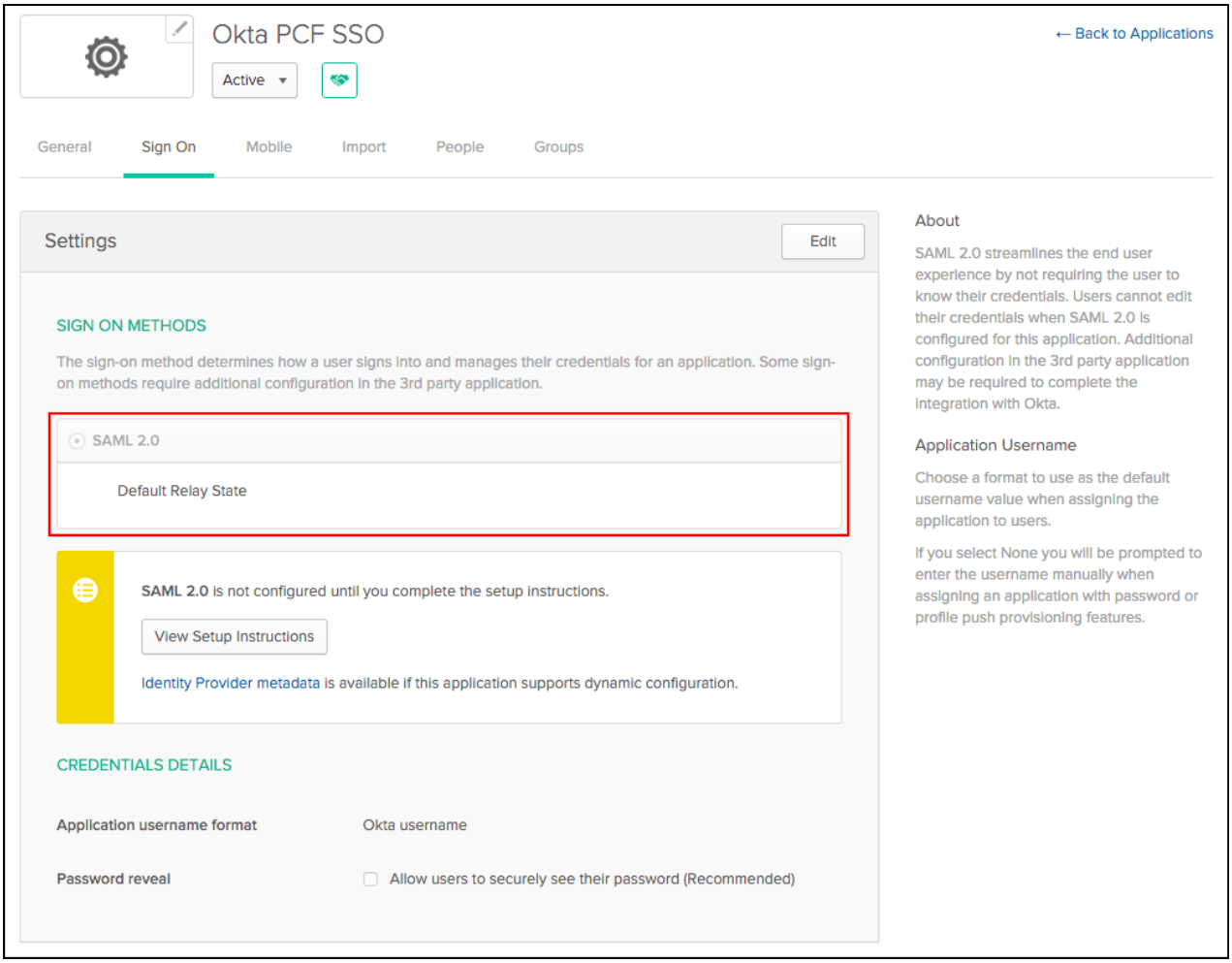
4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.
8. Open the downloaded service provider metadata file. You will refer to this file in the next step, when you fill in the SAML settings in Okta.

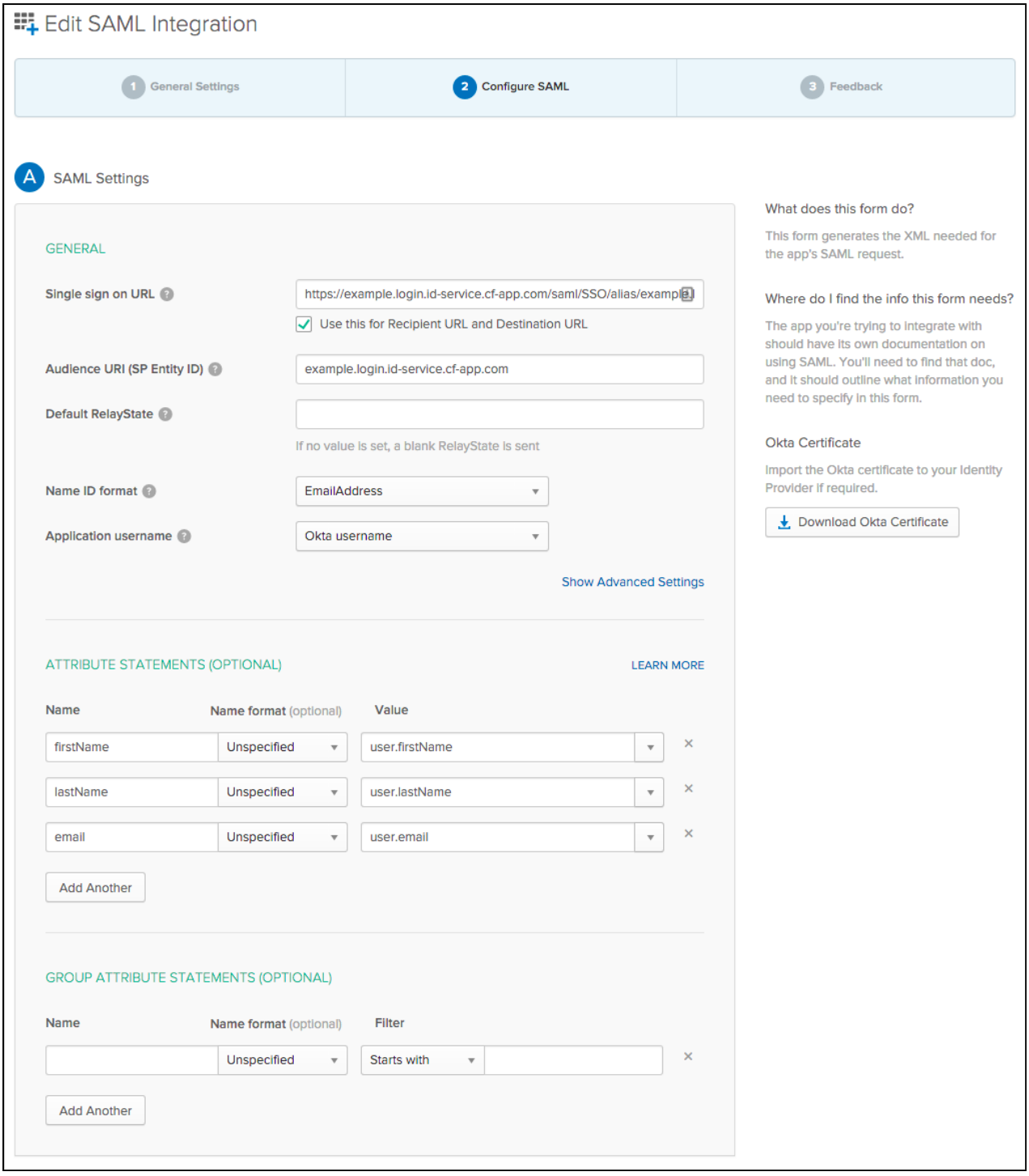
Set Up SAML in Okta

1. Sign in as an Okta administrator.
2. Navigate to your app and click the **Sign On** tab.
3. Under **Settings**, click **Edit**, and select **SAML 2.0**.



4. Click the **General** tab.

5. Under **SAML Settings**, click the **Edit** button followed by the **Next** button.



6. In the **SAML Settings** section:
    a. Enter the `AssertionConsumerService Location URL` from your downloaded service provider metadata into **Single sign on URL**. For example, `https://AUTH-DOMAIN/saml/SSO/alias/AUTH-DOMAIN`.
    b. Enter your Auth Domain URL into **Audience URI (SP Entity ID)**. You can view the Auth Domain for a plan by logging into the SSO dashboard, clicking the name of your plan, and selecting **Edit Plan**. For example, `https://AUTH-DOMAIN.login.SYSTEM-DOMAIN`. This value is also available in the downloaded service provider metadata as the entity ID.
    c. Select a **Name ID format**.
    d. Select an **Application username**.

7. (Optional) To configure single logout:
    a. Click **Show Advanced Settings**.
    b. For **Enable Single Logout**, select **Allow application** to initiate single logout.
    c. Enter the `SingleLogoutService Location URL` from your downloaded service provider metadata into **Single Logout URL**.

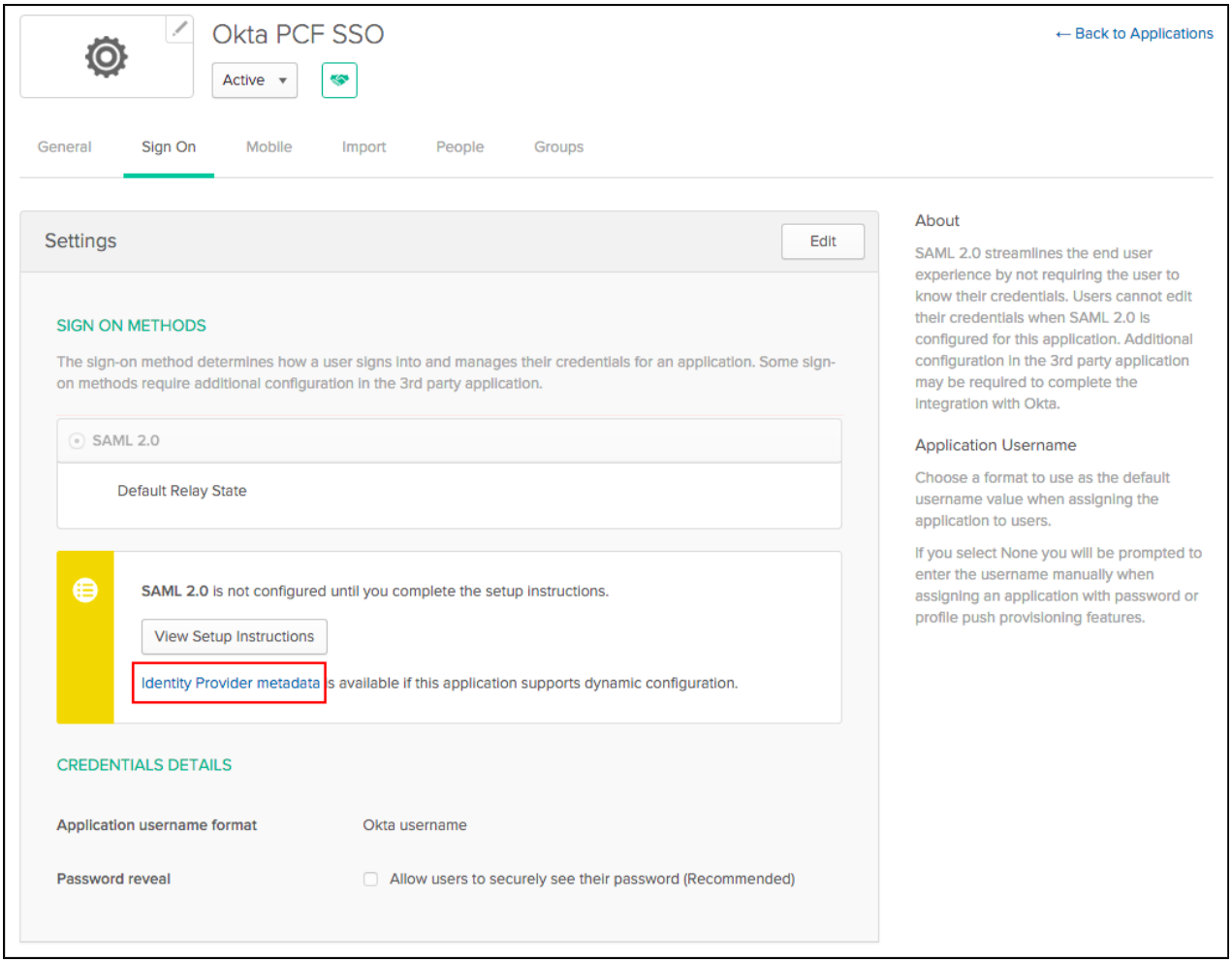
d. Enter your `Auth Domain URL` into **SP Issuer**.
e. Click **Upload Signature Certificate** to upload the signature certificate from your downloaded service provider metadata. You will need to copy the `X509Certificate` information from the downloaded service provider metadata, and reformat it into a valid certificate file to upload.

8. (Optional) Under **Attribute Statements (Optional)**, specify any attribute statements that you want to map to users in the ID token.

9. (Optional) Under **Group Attribute Statements (Optional)**, specify any group attribute statements that you want to map to users in the ID token. This is a group that users belong to within Okta.

10. Click the **Next** button followed by the **Finish** button.

11. Click **Identity Provider metadata** to download the metadata, or copy and save the link address of the **Identity Provider metadata**. You will need this Okta metadata for the next step, Configure a Single Sign-On Service Provider.



Configure a Single Sign-On Service Provider

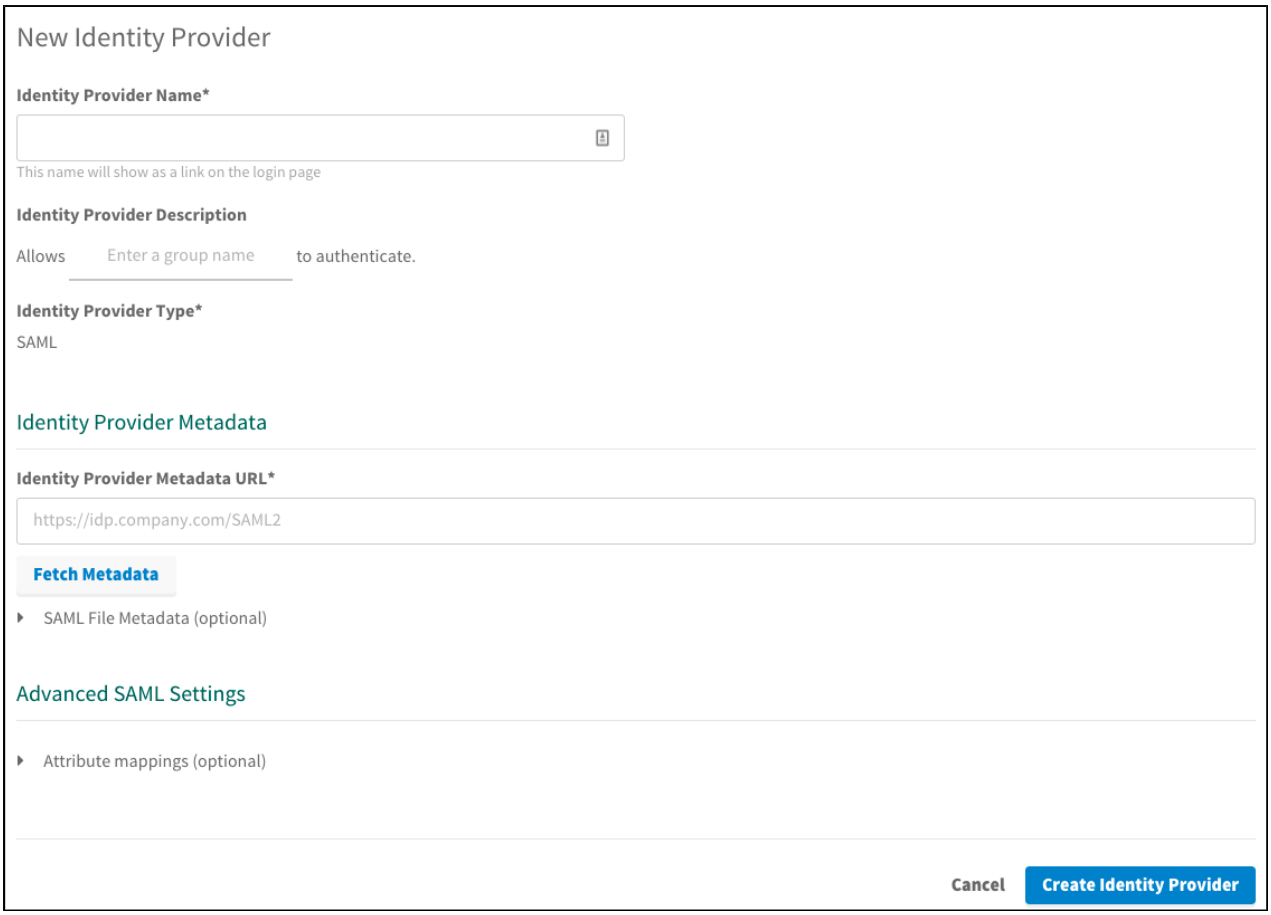
This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Setting up SAML

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **New Identity Provider** to create a new identity provider.



4. To create a new identity provider, perform the following steps:

    a. Enter an identity provider name into **Identity Provider Name**.
    b. (Optional) Enter a description into **Identity Provider Description**.
    c. Specify Identity Provider Metadata from Step 11 of the Configure Okta as an Identity Provider topic.

        i. Option 1: Enter your **Input Identity Provider Metadata URL** and **Fetch Metadata** to fetch your identity provider metadata from an endpoint.
        ii. Option 2: Click **SAML File Metadata (optional)** to upload your metadata XML manually.

    d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

6. Click **Resource Permissions**.

7. Click **New Permissions Mapping** and perform the following steps:

    a. Enter a **Group Name**.
    b. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.

8. Navigate to the identity provider list.

9. Click **Group Whitelist** and enter the group names from the external identity provider that should be propagated in the ID token.

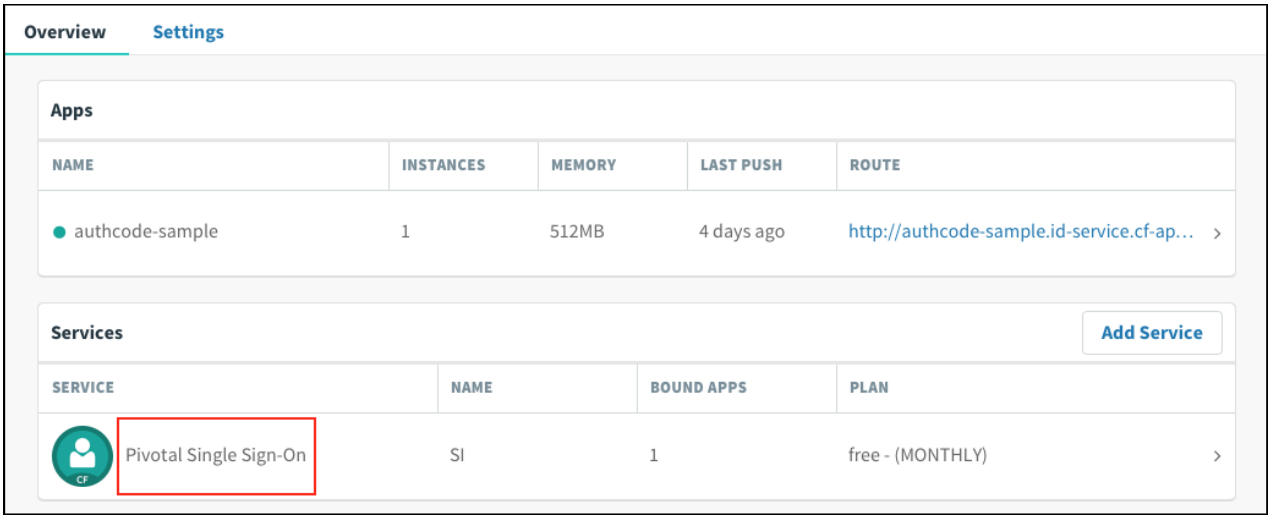
Testing

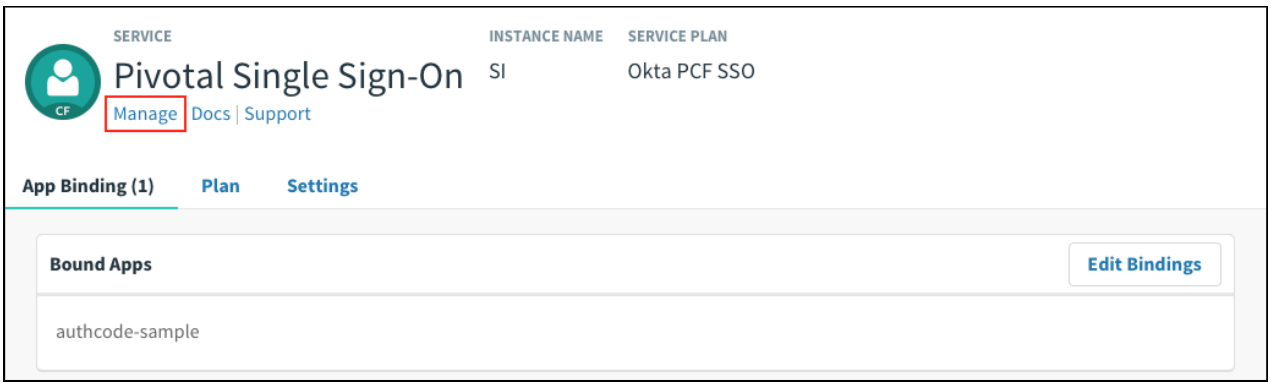
This topic describes how an administrator can test the connection between SSO and Okta services. An administrator can test both service provider and identity provider connections.

Test Your Service Provider Connection

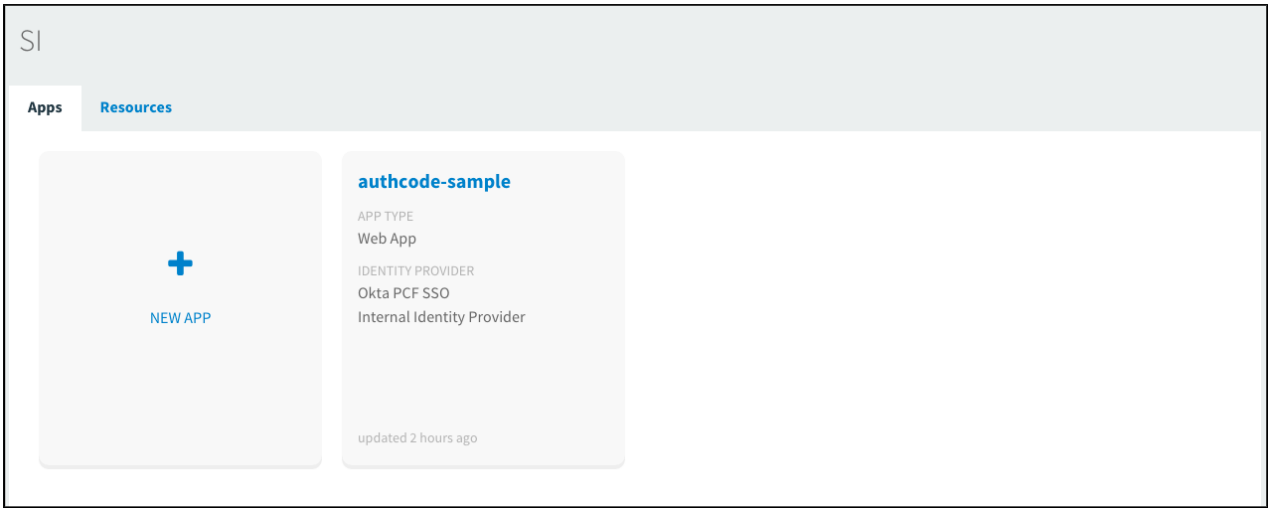
1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN` and navigate to the organization and space where your application is located.

2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application and click **Manage**.

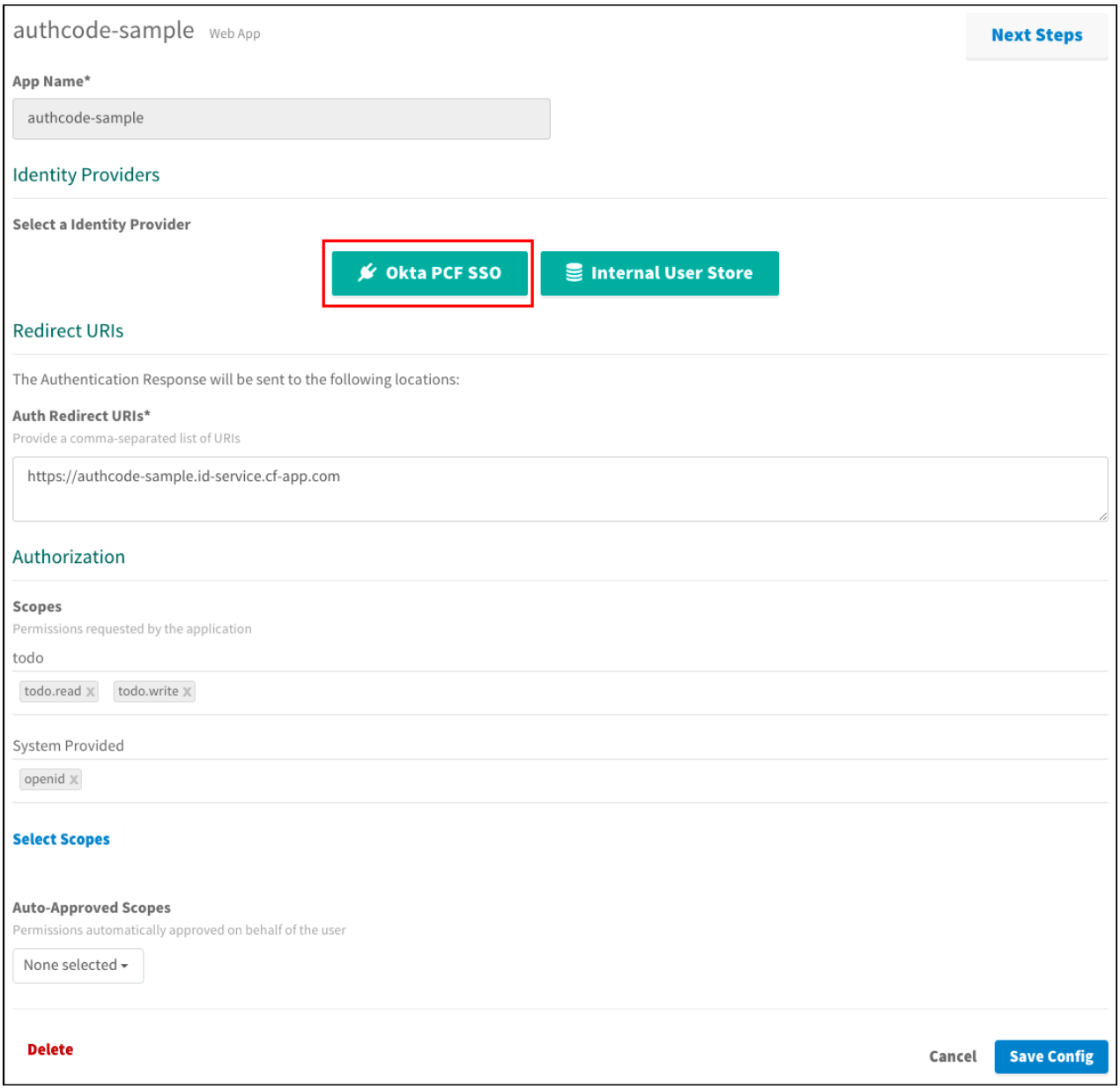




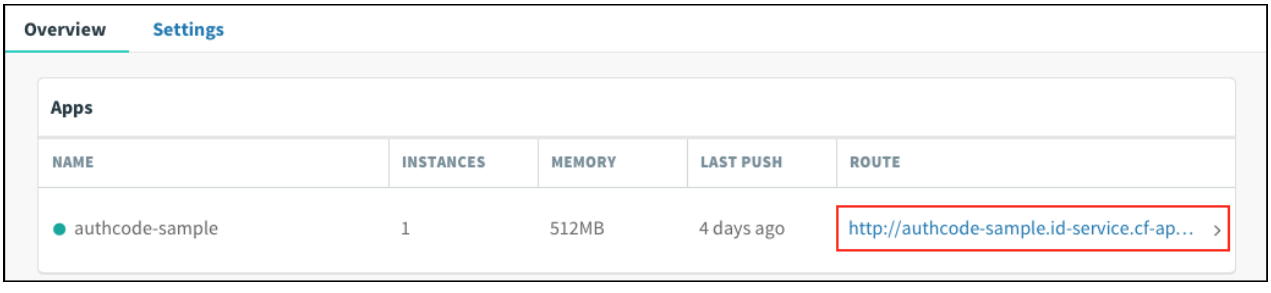
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the Okta identity provider.

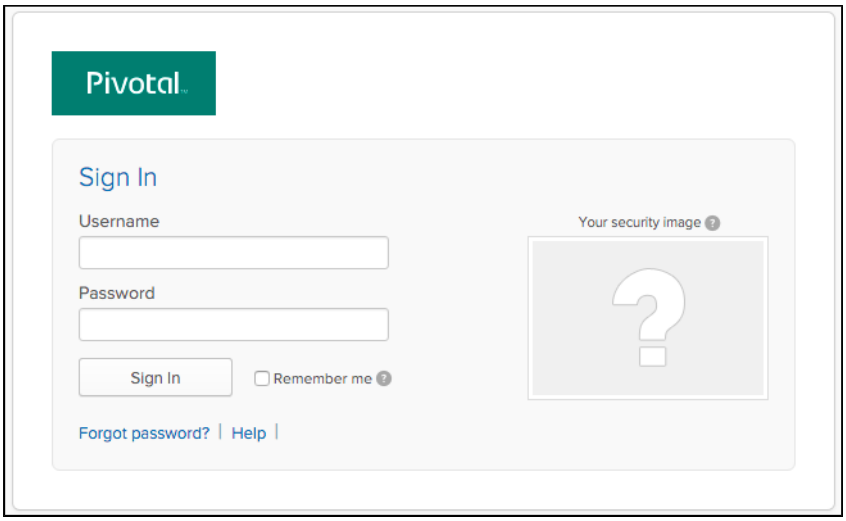


5. Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.

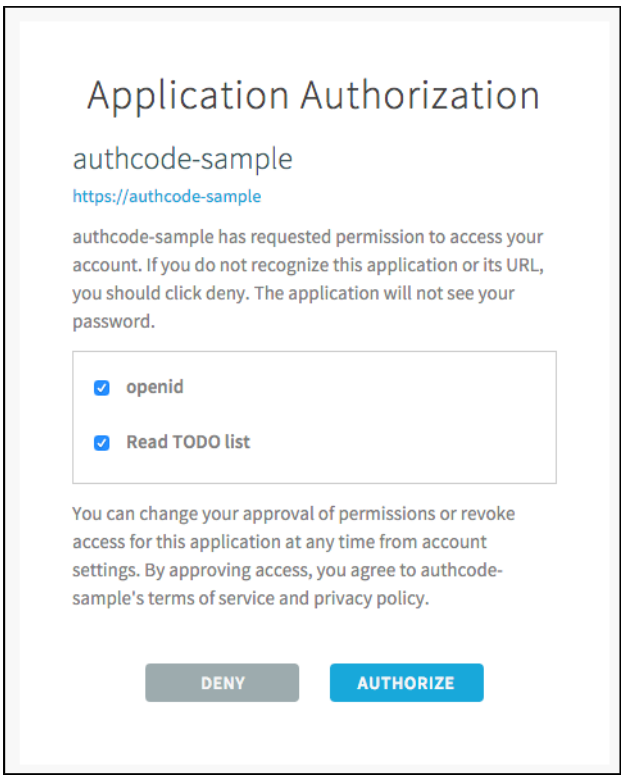


6. Click the link.

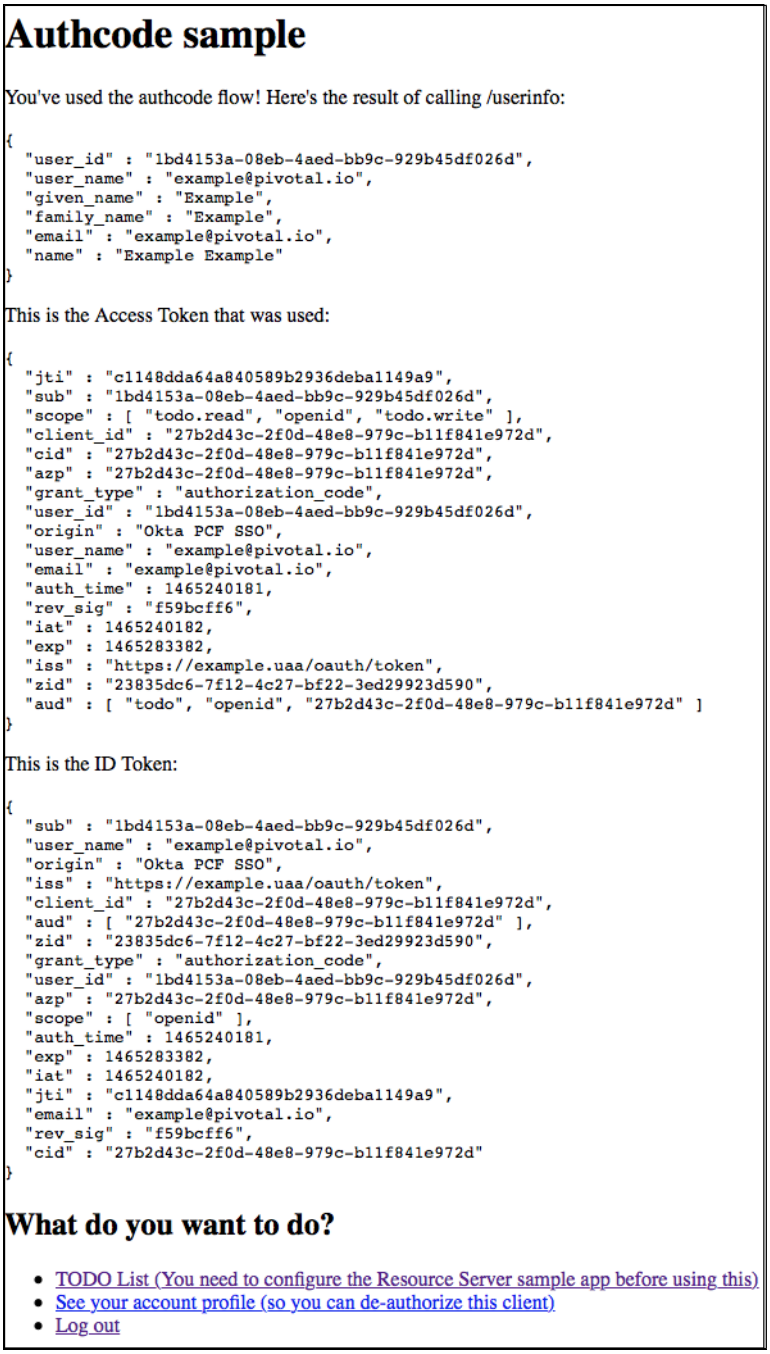
7. On the identity provider sign-in page, enter your credentials and click **Sign In**.



8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. The access token and ID token displays.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "user_name" : "example@pivotal.io",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "c1148dda64a840589b2936deba1149a9",
  "sub" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "cid" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "azp" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "grant_type" : "authorization_code",
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "origin" : "Okta PCF SSO",
  "user_name" : "example@pivotal.io",
  "email" : "example@pivotal.io",
  "auth_time" : 1465240181,
  "rev_sig" : "f59bcff6",
  "iat" : 1465240182,
  "exp" : 1465283382,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "23835dc6-7f12-4c27-bf22-3ed29923d590",
  "aud" : [ "todo", "openid", "27b2d43c-2f0d-48e8-979c-b11f841e972d" ]
}
```


This is the ID Token:


```
{
  "sub" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "user_name" : "example@pivotal.io",
  "origin" : "Okta PCF SSO",
  "iss" : "https://example.uaa/oauth/token",
  "client_id" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "aud" : [ "27b2d43c-2f0d-48e8-979c-b11f841e972d" ],
  "zid" : "23835dc6-7f12-4c27-bf22-3ed29923d590",
  "grant_type" : "authorization_code",
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "azp" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "scope" : [ "openid" ],
  "auth_time" : 1465240181,
  "exp" : 1465283382,
  "iat" : 1465240182,
  "jti" : "c1148dda64a840589b2936deba1149a9",
  "email" : "example@pivotal.io",
  "rev_sig" : "f59bcff6",
  "cid" : "27b2d43c-2f0d-48e8-979c-b11f841e972d"
}
```

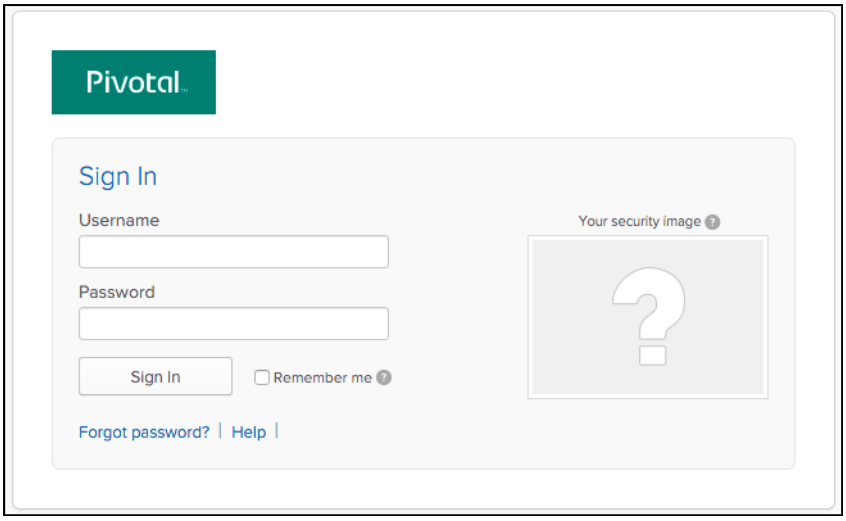

What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

Test Your Identity Provider Connection

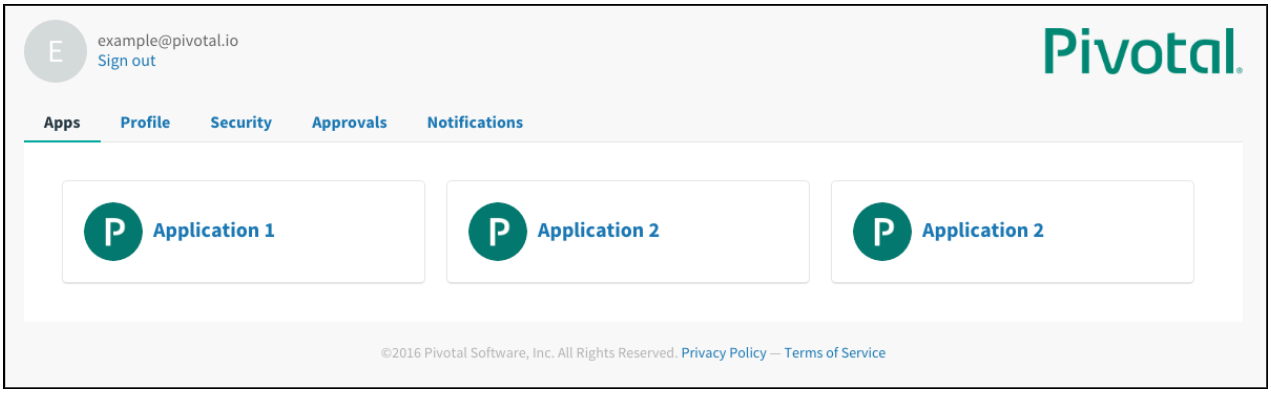
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign into Okta.



2. Navigate to the application tile and click it.

3. You are redirected to the page that lists applications you have access to.

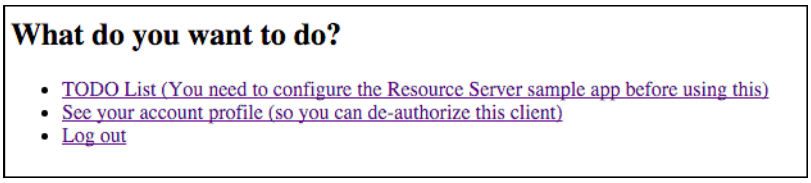


Test Your Single Sign-Off

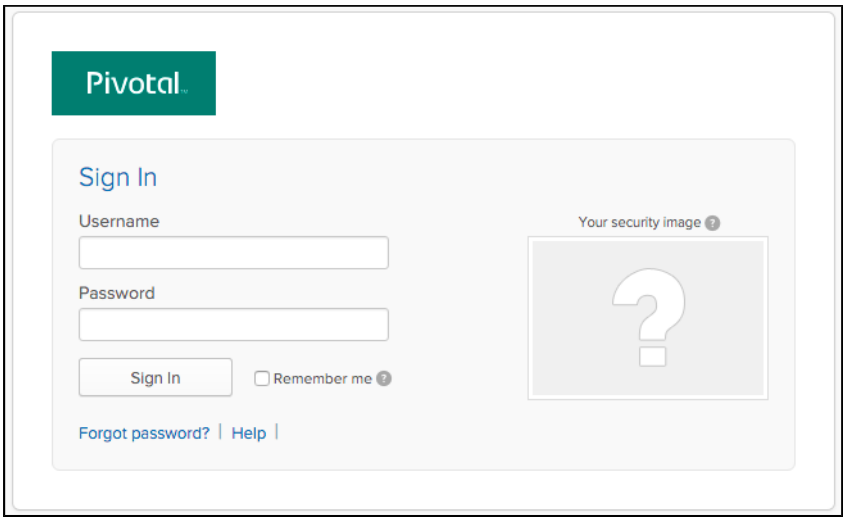
Test single sign-off to ensure that when users log out of the application, they are logged out of Okta as well.

1. Sign into the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

2. Under “What do you want to do?”, click **Log out**.



3. You are logged out and redirected to the Okta login page.

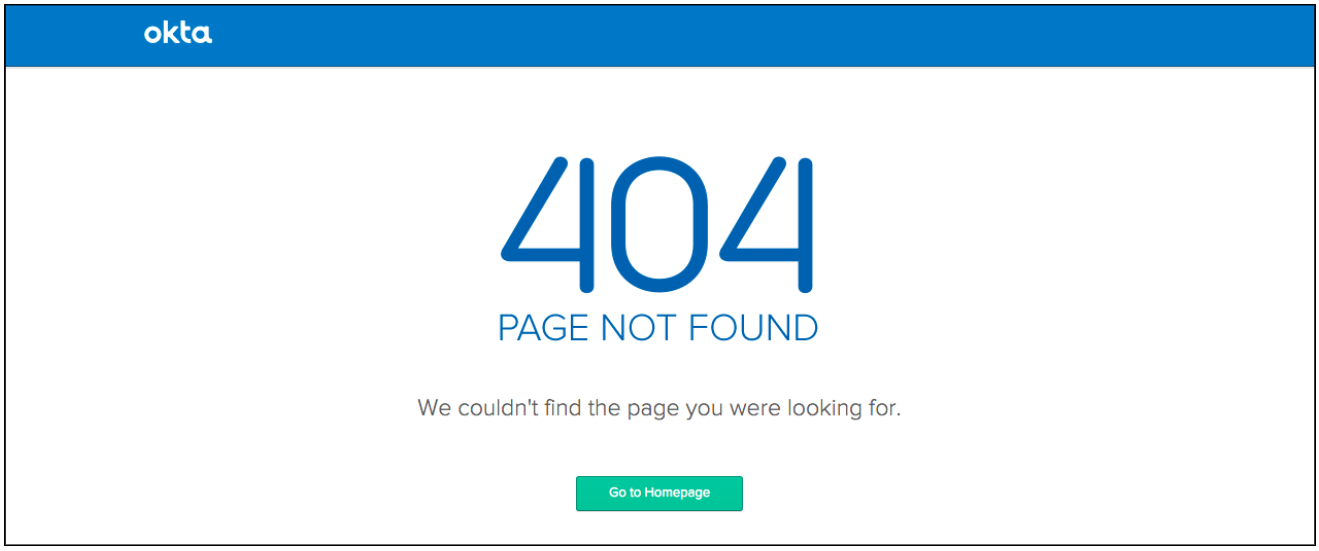


Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between Okta and Pivotal Single Sign-On (SSO).

Page Not Found

Symptom:

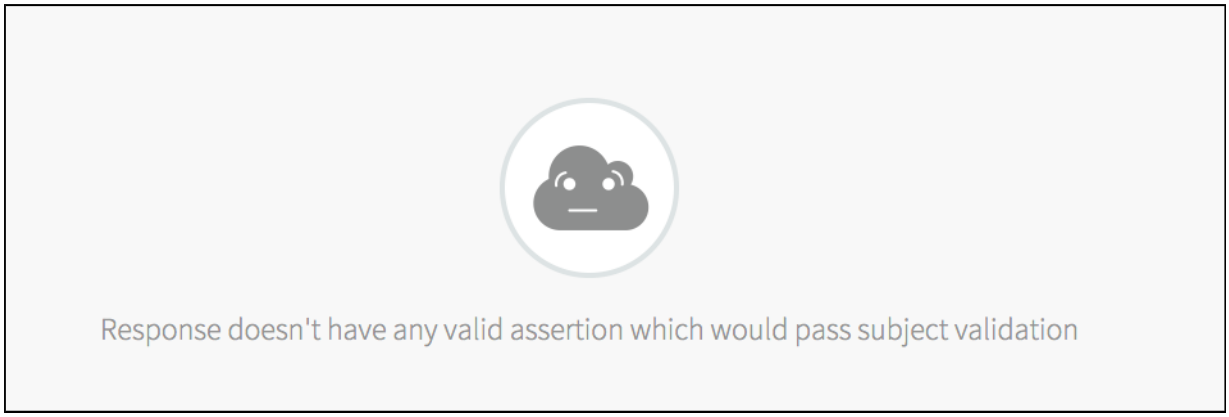


Explanations:

- The Okta instance is inactive.
- The Recipient URL is misconfigured in Okta.
- The identity provider SSO URL is misconfigured in the SSO plan settings.

No Valid Assertion

Symptom:

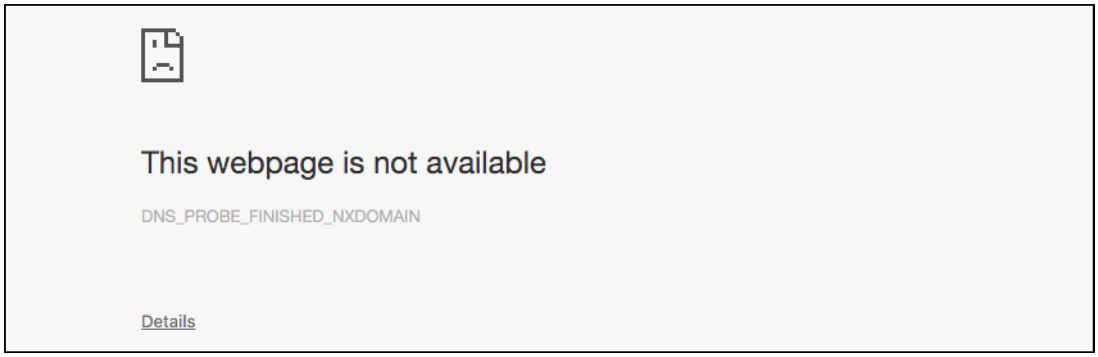


Explanations:

- The service provider Entity ID is misconfigured in Okta.
- The Destination URL is misconfigured in Okta.

Webpage Not Available

Symptom:

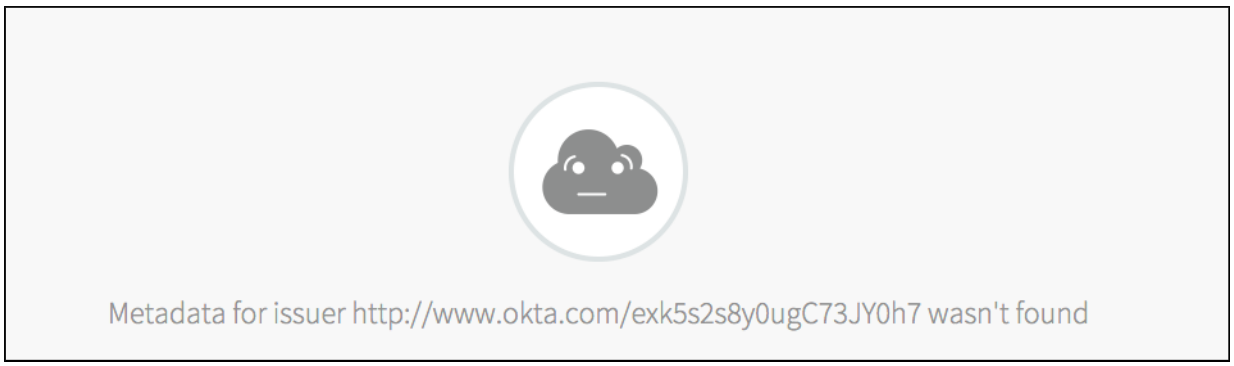


Explanation:

- The SSO URL is misconfigured in Okta.

Metadata Not Found

Symptom:



Explanation:

- The identity provider Entity ID is misconfigured in the SSO plan settings.

PingFederate Integration Guide Overview

PingFederate is a federation server that provides identity management, single sign-on, and API security for the enterprise. This documentation describes how to configure a single sign-on partnership between PingFederate as the Identity Provider (IdP) and the Single Sign-On Service (SSO) for Pivotal Cloud Foundry as the Service Provider (SP).

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

Prerequisites

To integrate PingFederate with Pivotal Cloud Foundry (PCF), you need:

Pivotal

- PCF, version 1.7.0 or later.
- Single Sign-On, version 1.1.0 or later.

Ping

- PingFederate
- A user with Administrator privileges.

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic.

PingFederate Integration Guide

Configuring PingFederate with SSO

Complete both steps below to integrate your deployment with PingFederate and SSO.

1. Configure PingFederate as an Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

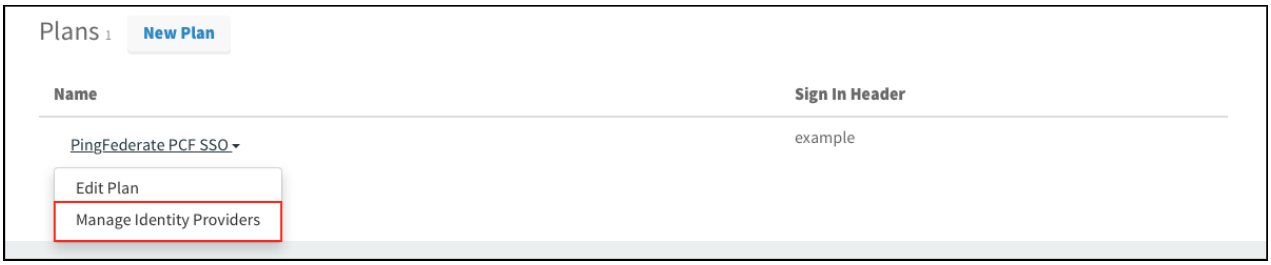
- Testing
- Troubleshooting

Configure PingFederate as an Identity Provider

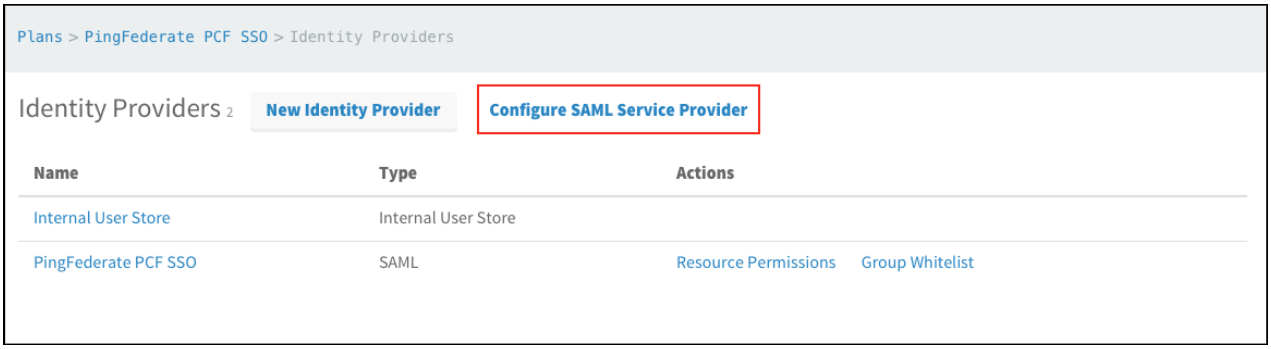
This topic describes how to set up PingFederate as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and PingFederate.

Set up SAML in PCF

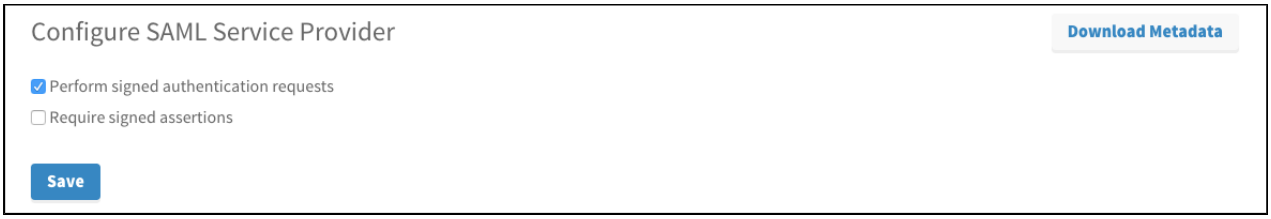
1. Log into the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.
2. Select your plan and choose **Manage Identity Providers** from the drop-down menu.



3. Click **Configure SAML Service Provider**.



4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.

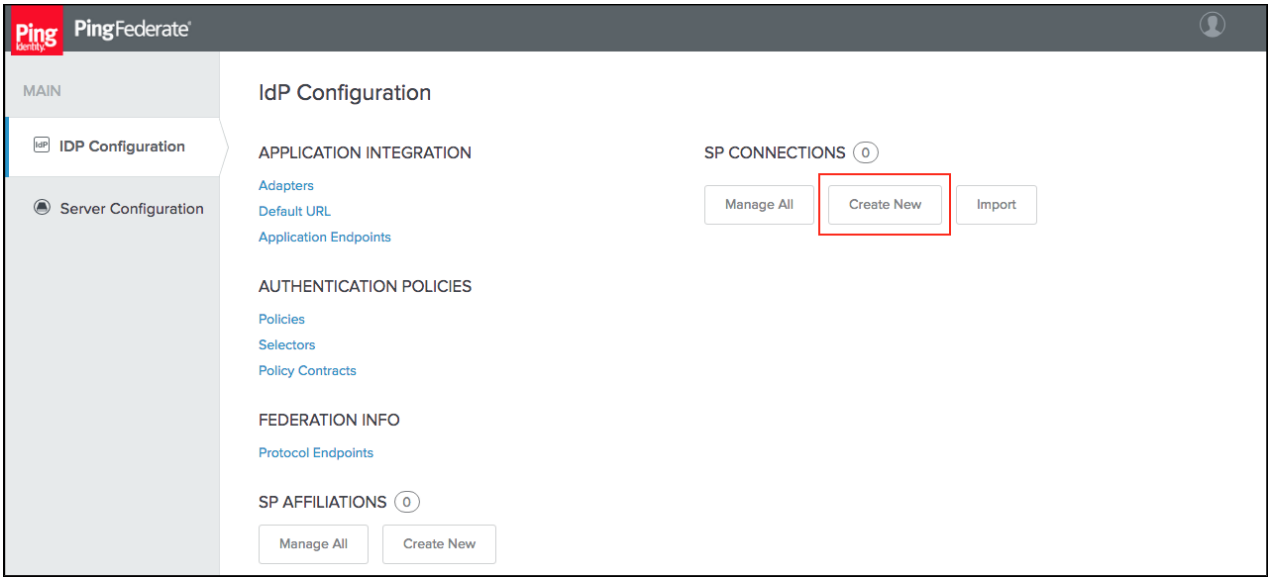


5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

Set up SAML in PingFederate

Configure the Connection

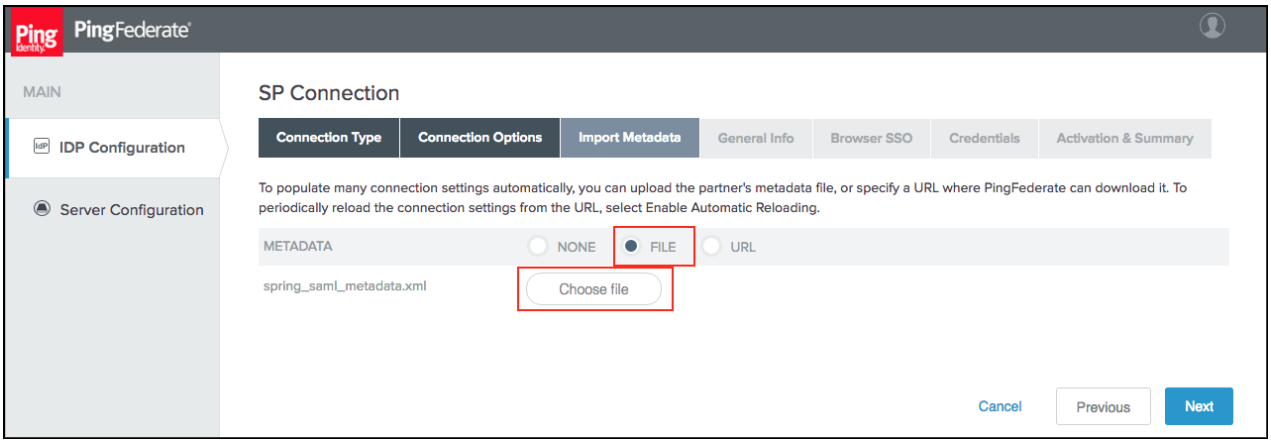
1. Sign in as a PingFederate administrator.
2. Navigate to your identity provider configurations by clicking on the **IDP Configuration** tab.
3. Under **SP Connections**, click the **Create New** button.



4. Select the **Browser SSO Profiles** connection template on the **Connection Type** tab and click **Next**.

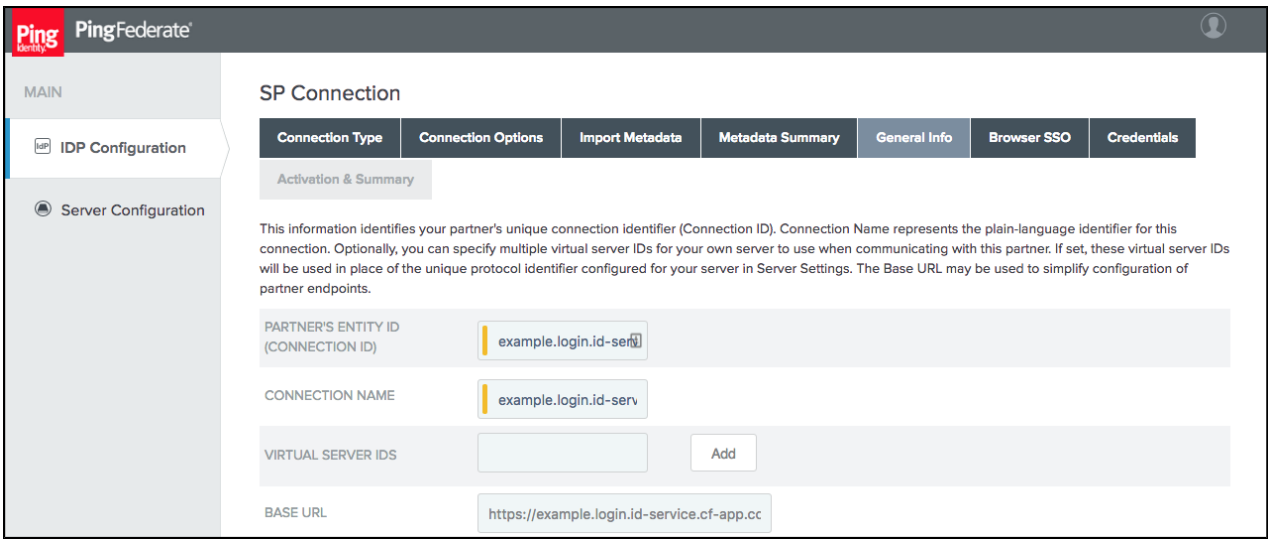
5. Select **Browser SSO** on the **Connection Options** tab and click **Next**.

6. Select **File** as the method for importing metadata and click **Choose file** to choose the SSO metadata on the **Import Metadata** tab. Click **Next**.



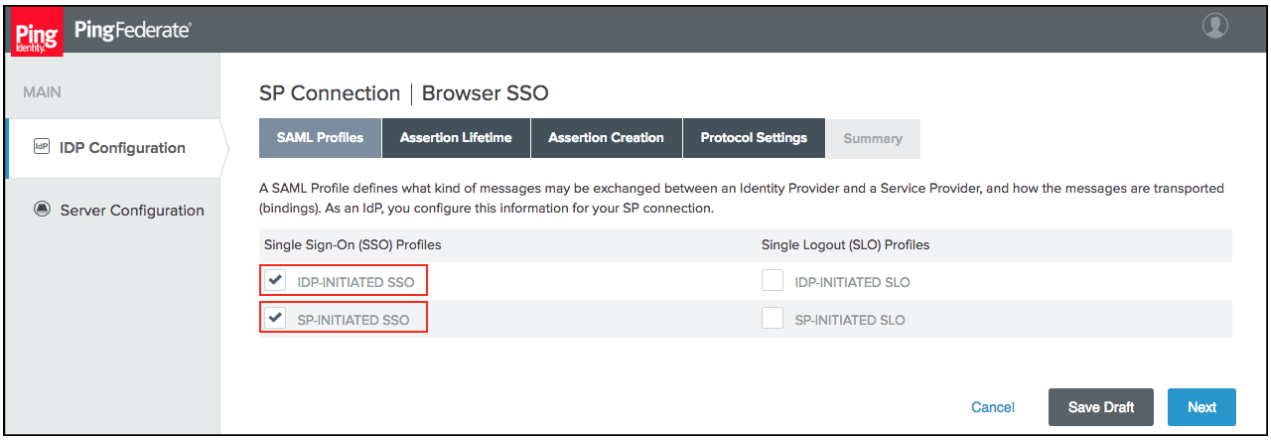
7. Review the information on the **Metadata Summary** tab and click **Next**.

8. Ensure that the **Partner's Entity ID**, **Connection Name**, and **Base URL** fields pre-populate based on the metadata. Click **Next**.



Configure Browser SSO

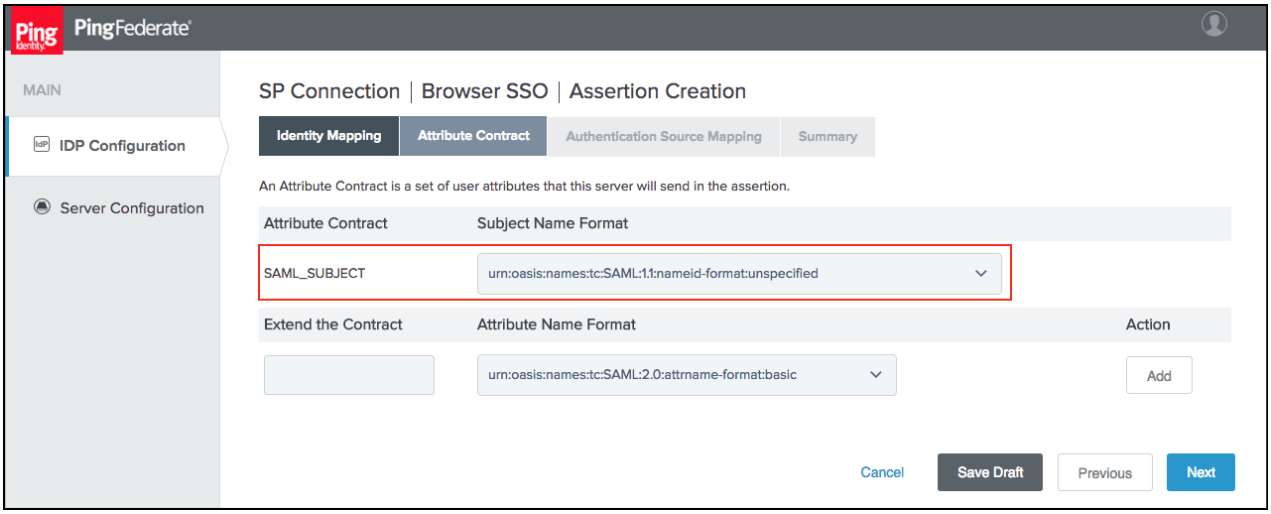
1. Click **Configure Browser SSO** on the **Browser SSO** tab.
2. Select the **IdP-Initiated SSO** and **SP-Initiated SSO** options on the **SAML Profiles** tab and click **Next**.



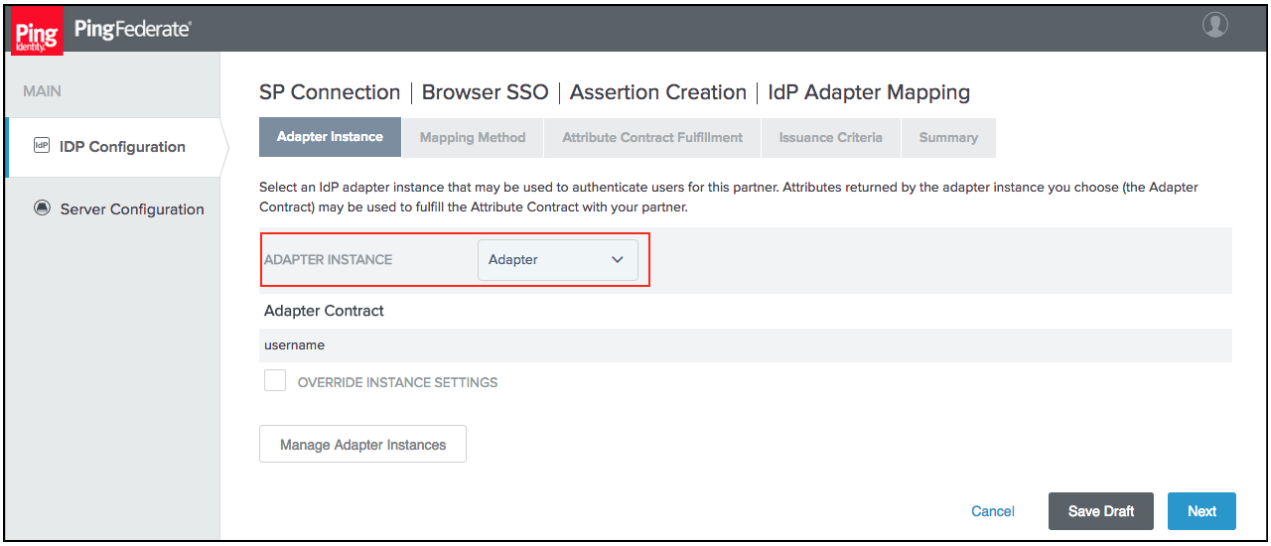
3. Enter your desired assertion validity time from on the **Assertion Lifetime** tab and click **Next**.
4. (Optional) Select **IdP-Initiated SLO** and **SP-Initiated SLO** options if you wish to enforce Single Logout.

Assertion Creation

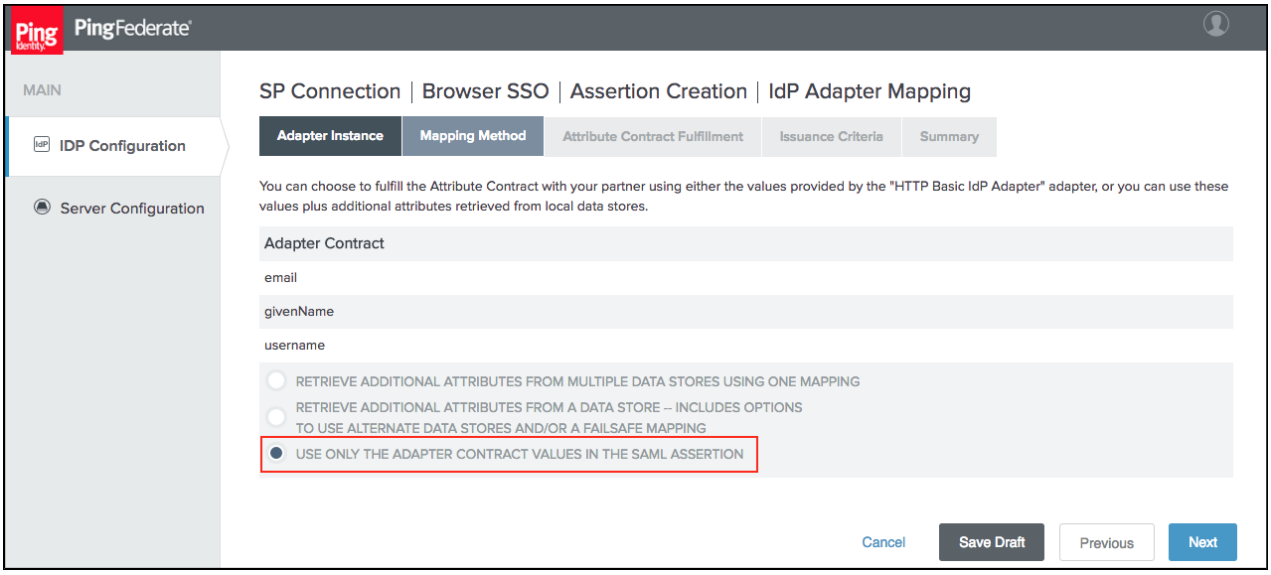
1. Click **Configure Assertion Creation** on the **Assertion Creation** tab.
2. Choose the **Standard** option on the **Identity Mapping** tab and click **Next**.
3. Select a **Subject Name Format** for the **SAML_SUBJECT** on the **Attribute Contract** tab and click **Next**.



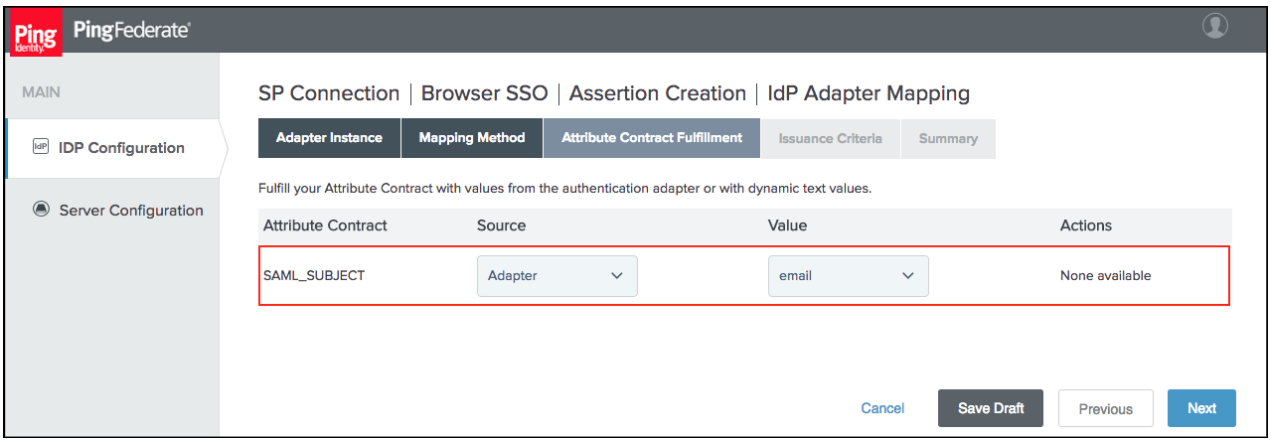
4. Click **Map New Adapter Instance** on the **Authentication Source Mapping** tab.
5. Select an **Adapter Instance** and click **Next**. The adapter must include the user's email address.



6. Select the **Use only the adapter contract values in the SAML assertion** option on the **Mapping Method** tab and click **Next**.



7. Select your adapter instance as the **Source** and the email as the **Value** on the **Attribute Contract Fullfillment** tab and click **Next**.

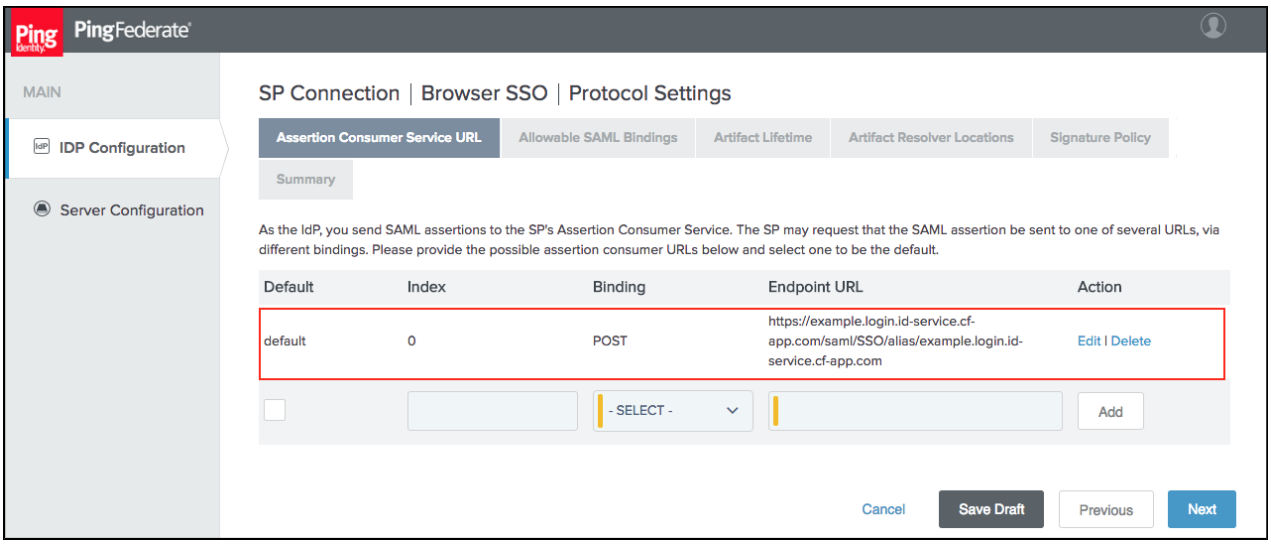


8. (Optional) Select any authorization conditions you would like on the **Issuance Criteria** tab and click **Next**.
9. Click **Done** on the **Summary** tab.
10. Click **Next** on the **Authentication Source Mapping** tab.
11. Click **Done** on the **Summary** tab.

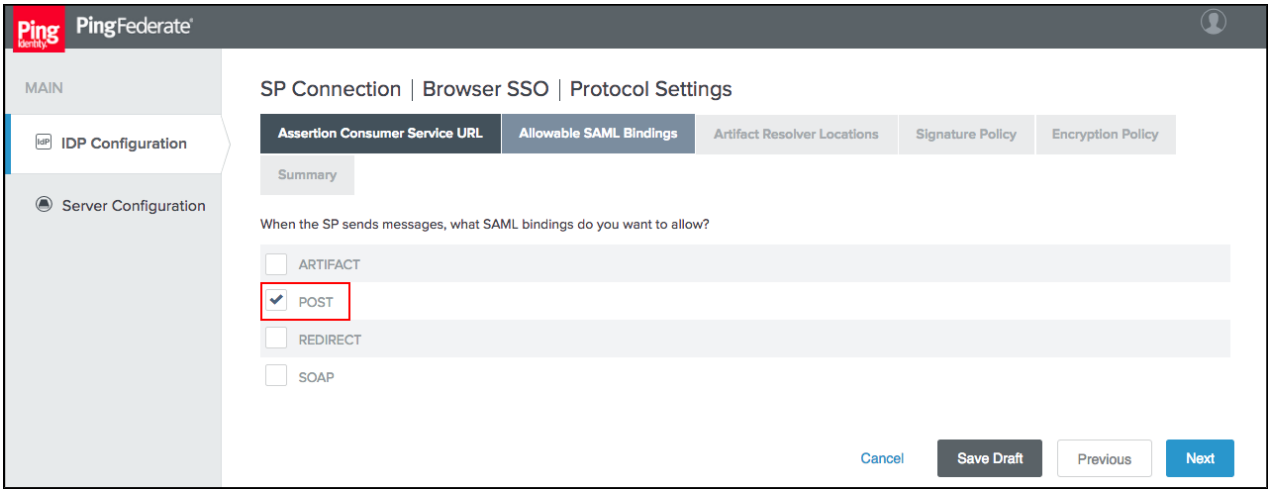
12. Click **Next** on the **Assertion Creation** tab.

Protocol Settings

1. Click **Configure Protocol Settings** on the **Protocol Settings** tab.
2. Select POST for **Binding** and specify the single sign-on endpoint url in the **Endpoint URL** field on the **Assertion Consumer Service URL** tab. Click **Next**



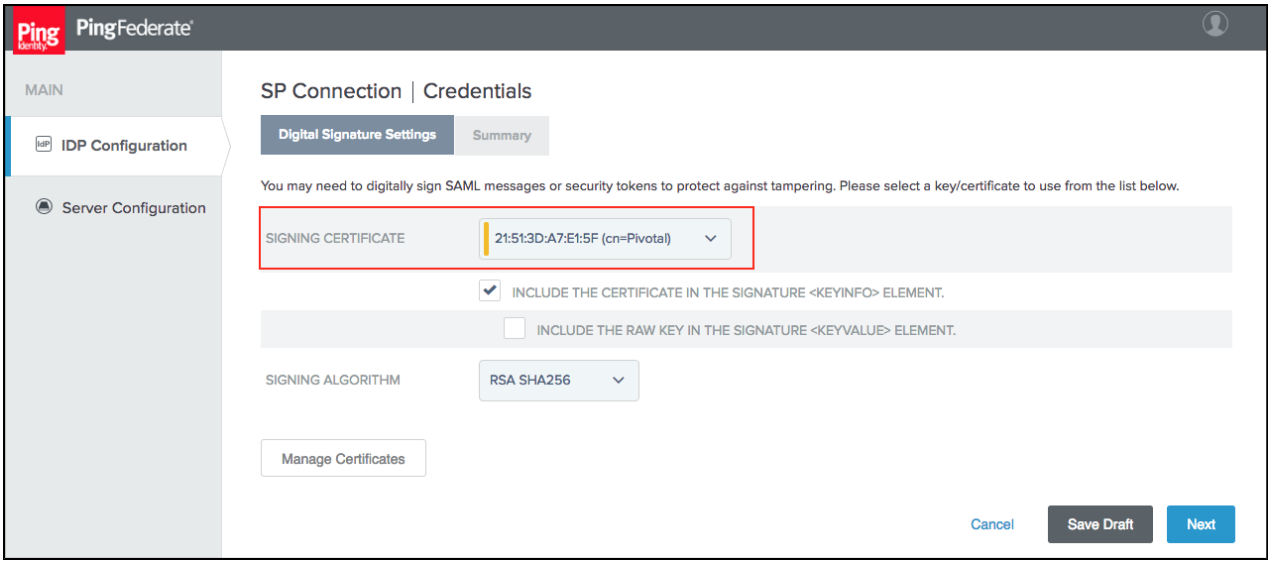
3. Select **POST** on the **Allowable SAML Bindings** tab and click **Next**.



4. Select your desired signature policies for assertions on the **Signature Policy** tab and click **Next**.
5. Select your desired encryption policy for assertions on the **Encryption Policy** tab and click **Next**.
6. Click **Done** on the **Protocol Settings Summary** tab.
7. Click **Done** on the **Browser SSO Summary** tab.

Configure Credentials

1. Click **Configure Credentials** on the **Credentials** tab.
2. Select the **Signing Certificate** to use with the Single Sign-On service and select **Include the certificate in the signature element**. Click **Next**.



3. Click **Done** on the **Summary** tab.
4. Click **Next** on the **Credentials** tab.
5. Select **Active** for the **Connection Status** on the **Activation & Summary** tab and click **Save**.
6. Click **Manage All** under **SP Connections**.
7. Click **Export Metadata** for the desired service provider connection.
8. Choose a **Signing Certificate** on the **Metadata Signing** tab and click **Next**.
9. Click **Export** on the **Export & Summary** tab and click **Done**.

Configure a Single Sign-On Service Provider

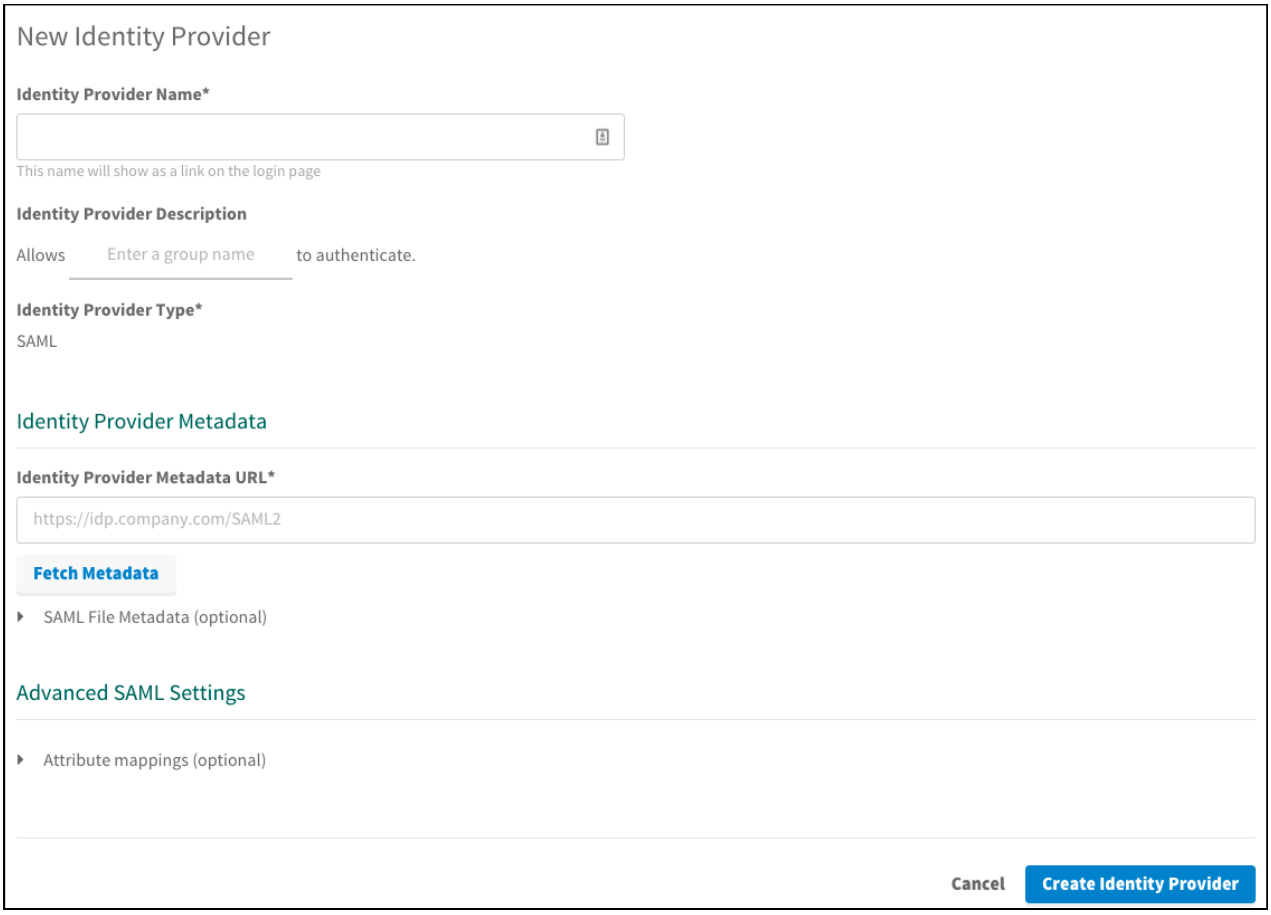
This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Setting up SAML

1. Log in to the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and choose **Manage Identity Providers** from the drop-down menu.

3. Click **New Identity Provider**.



4. To create a new identity provider, perform the following steps:

    a. Enter an identity provider name into **Identity Provider Name**.
    b. (Optional) Enter a description into **Identity Provider Description**.
    c. Click **SAML File Metadata (optional)**, then click the **Upload Identity Provider Metadata** button to upload your metadata XML.
    d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

6. Click **Resource Permissions**.

7. Click **New Permissions Mapping** and perform the following steps:
    a. Enter a **Group Name**.
    b. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.
8. Navigate to the identity provider list.
9. Click **Group Whitelist** and enter the group names from the external identity provider to propagate in the ID token when a user authenticates.

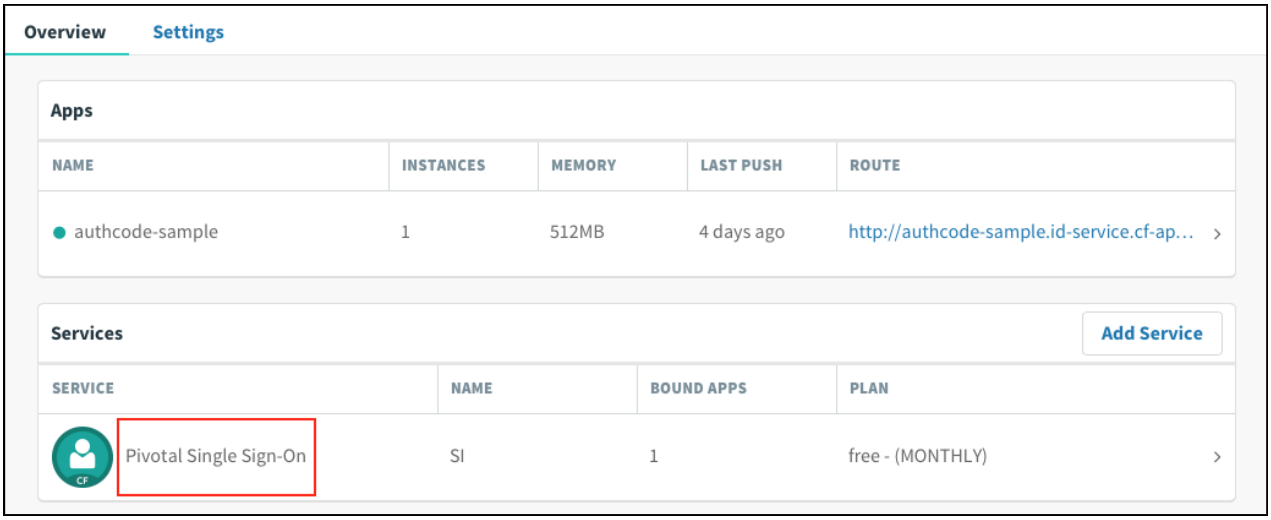
Testing

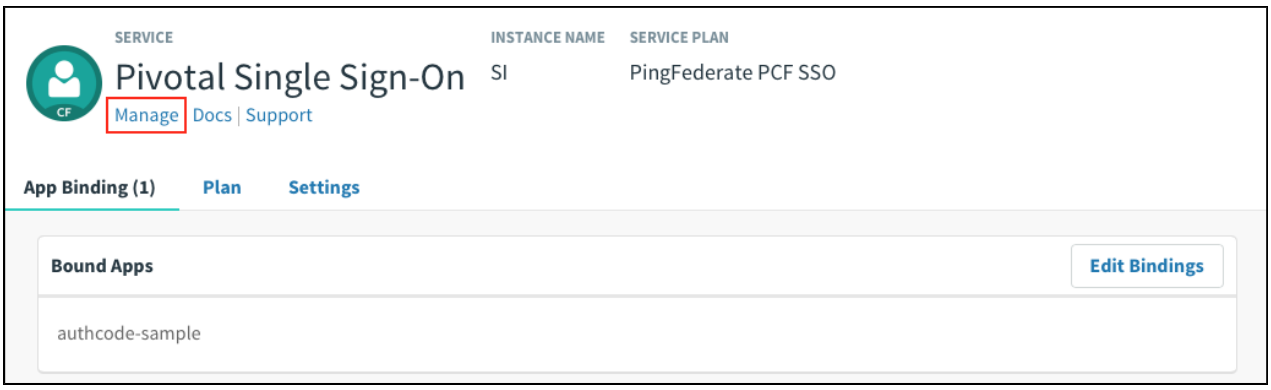
This topic describes how an administrator can test the connection between SSO and PingFederate. An administrator can test both service provider and identity provider connections.

Test Your Service Provider Connection

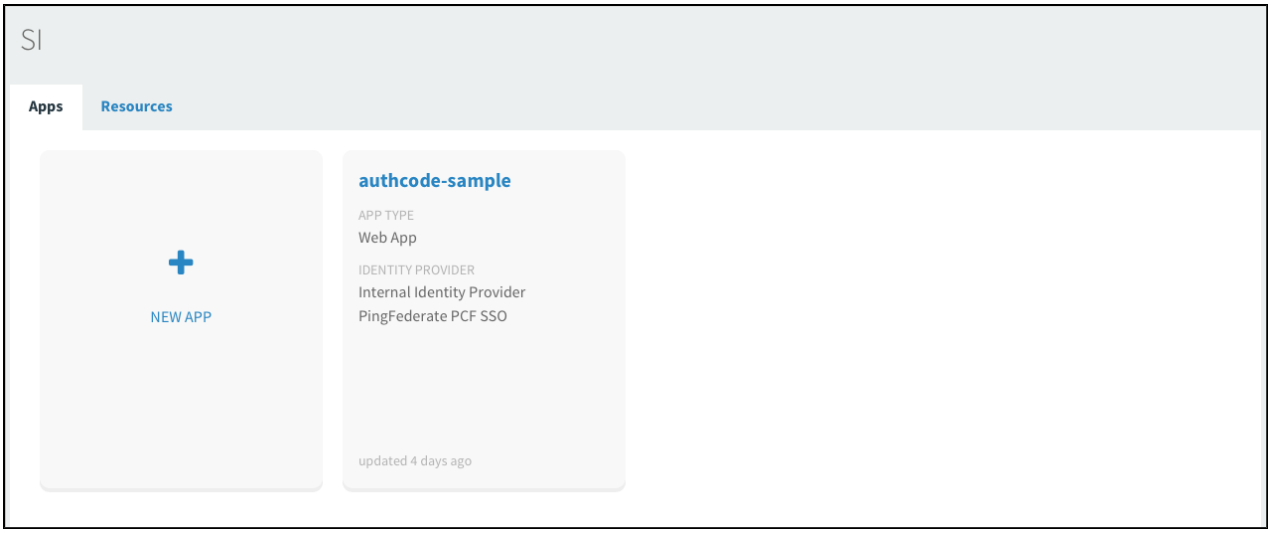
1. Log in to Apps Manager at https://apps.YOUR-SYSTEM-DOMAIN and navigate to the organization and space where your application is located.

2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Click the service instance and then click **Manage**.

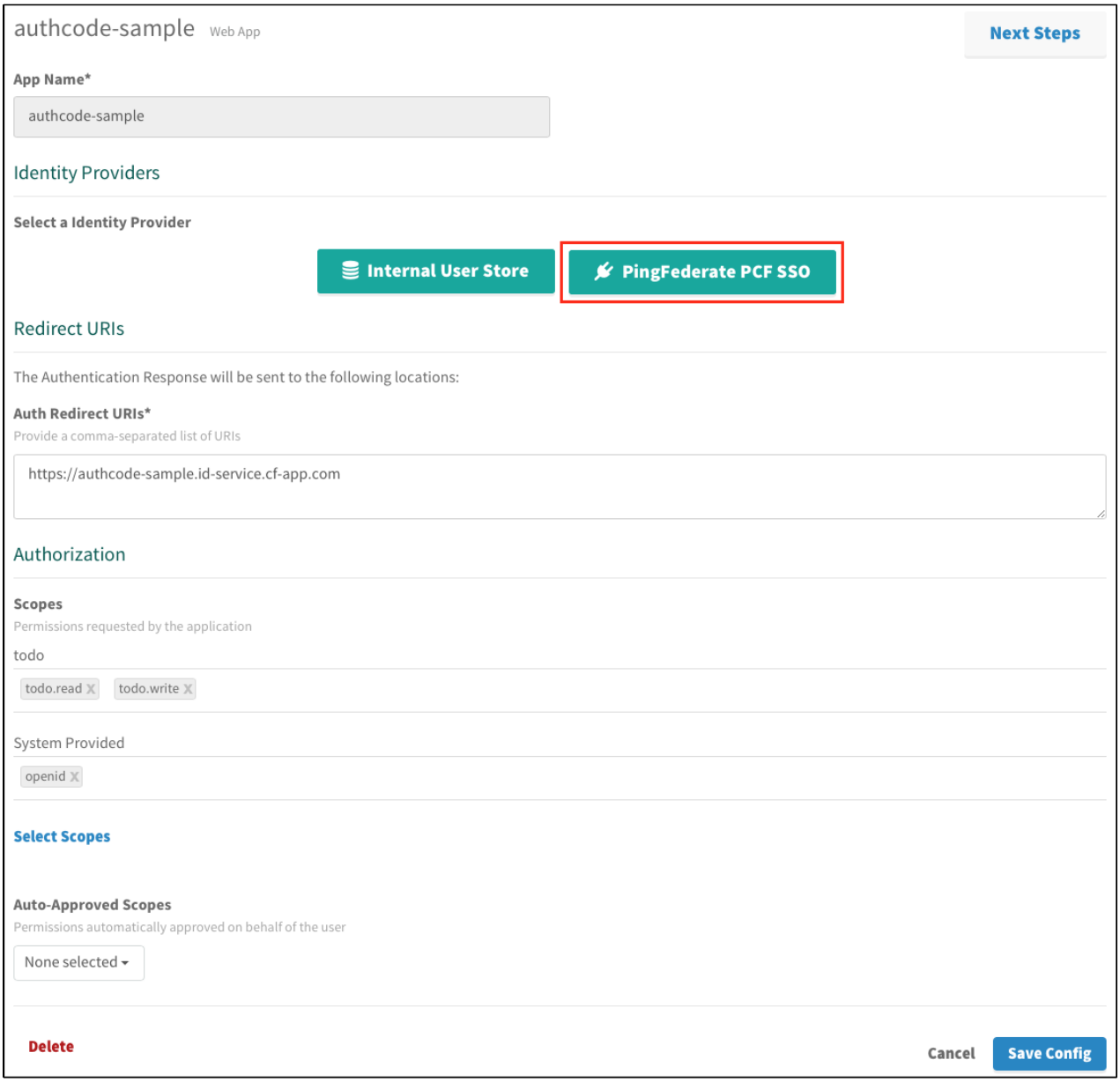




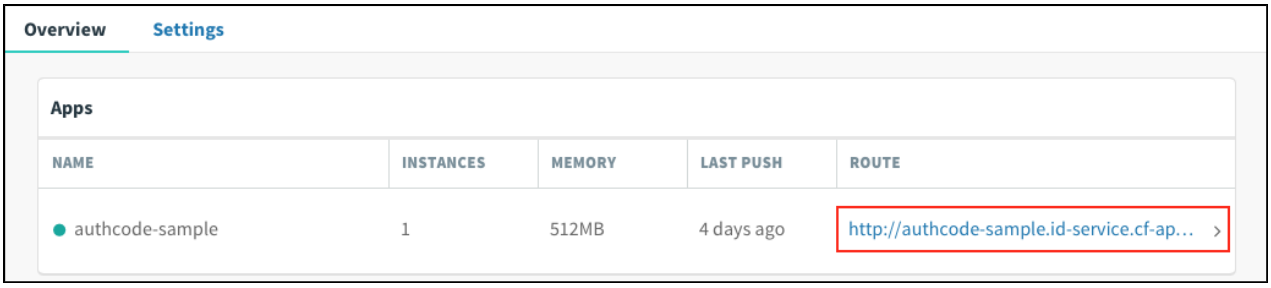
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the PingFederate identity provider. a

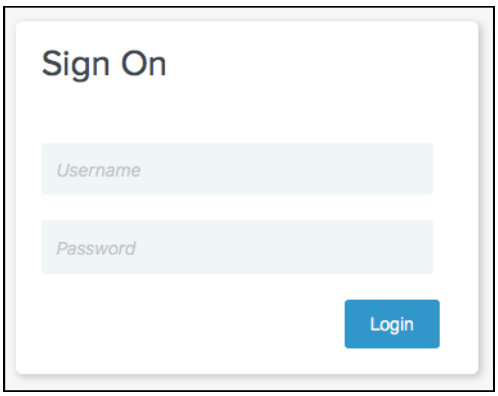


5. Return to Apps Manager and click the URL below your application to authenticate with the identity provider.

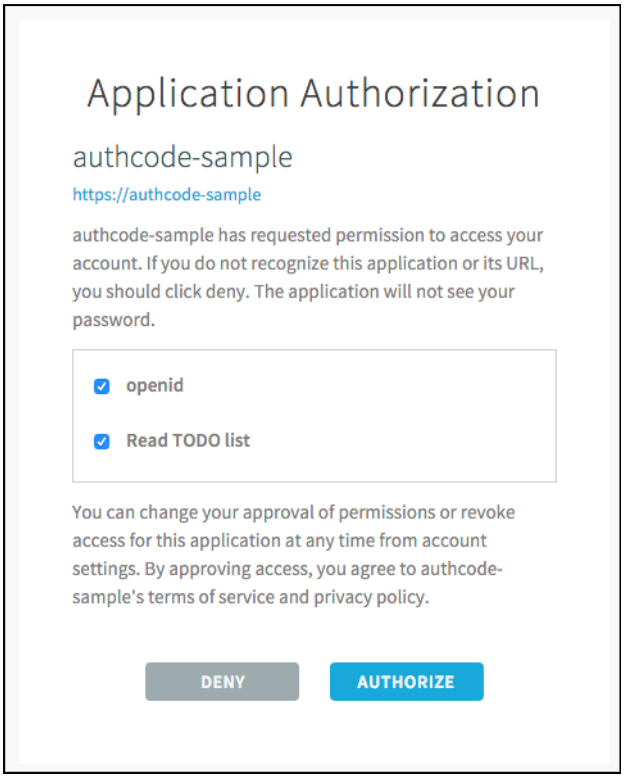


6. Click the link to **Log in via Auth Code Grant Type**.

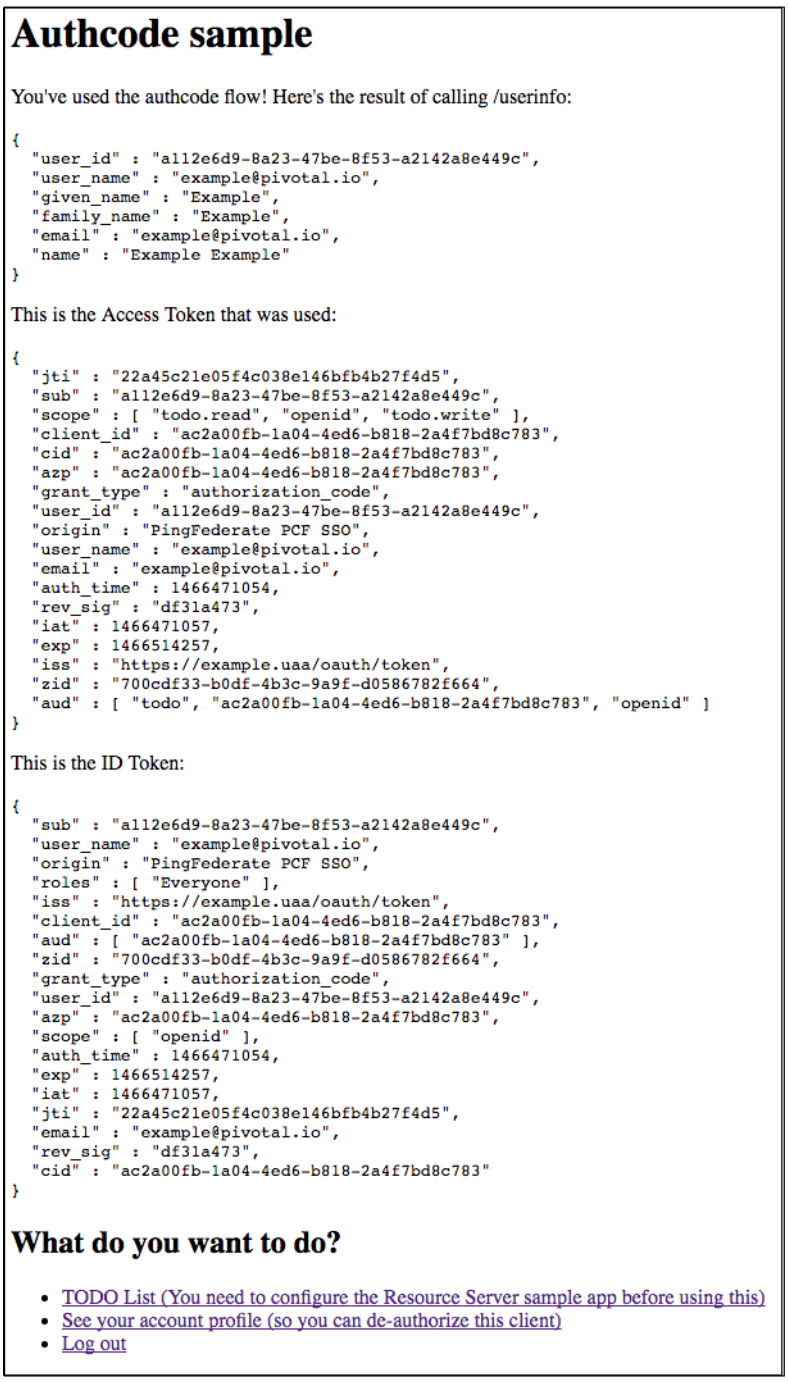
7. On the identity provider sign-in page, enter your credentials and click **Sign On**.



8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. View the access token and ID token.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "a112e6d9-8a23-47be-8f53-a2142a8e449c",
  "user_name" : "example@pivotal.io",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "22a45c21e05f4c038e146bfb4b27f4d5",
  "sub" : "a112e6d9-8a23-47be-8f53-a2142a8e449c",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783",
  "cid" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783",
  "azp" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783",
  "grant_type" : "authorization_code",
  "user_id" : "a112e6d9-8a23-47be-8f53-a2142a8e449c",
  "origin" : "PingFederate PCF SSO",
  "user_name" : "example@pivotal.io",
  "email" : "example@pivotal.io",
  "auth_time" : 1466471054,
  "rev_sig" : "df31a473",
  "iat" : 1466471057,
  "exp" : 1466514257,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "700cdf33-b0df-4b3c-9a9f-d0586782f664",
  "aud" : [ "todo", "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783", "openid" ]
}
```


This is the ID Token:


```
{
  "sub" : "a112e6d9-8a23-47be-8f53-a2142a8e449c",
  "user_name" : "example@pivotal.io",
  "origin" : "PingFederate PCF SSO",
  "roles" : [ "Everyone" ],
  "iss" : "https://example.uaa/oauth/token",
  "client_id" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783",
  "aud" : [ "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783" ],
  "zid" : "700cdf33-b0df-4b3c-9a9f-d0586782f664",
  "grant_type" : "authorization_code",
  "user_id" : "a112e6d9-8a23-47be-8f53-a2142a8e449c",
  "azp" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783",
  "scope" : [ "openid" ],
  "auth_time" : 1466471054,
  "exp" : 1466514257,
  "iat" : 1466471057,
  "jti" : "22a45c21e05f4c038e146bfb4b27f4d5",
  "email" : "example@pivotal.io",
  "rev_sig" : "df31a473",
  "cid" : "ac2a00fb-1a04-4ed6-b818-2a4f7bd8c783"
}
```

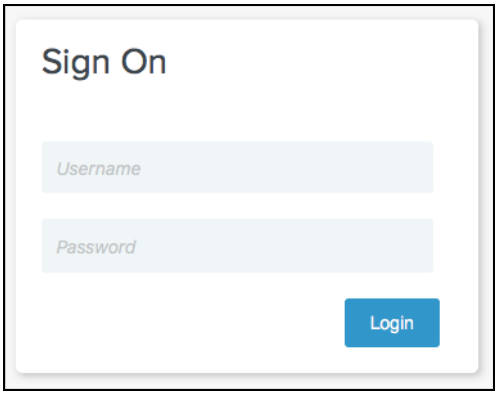

What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

Test Your Identity Provider Connection

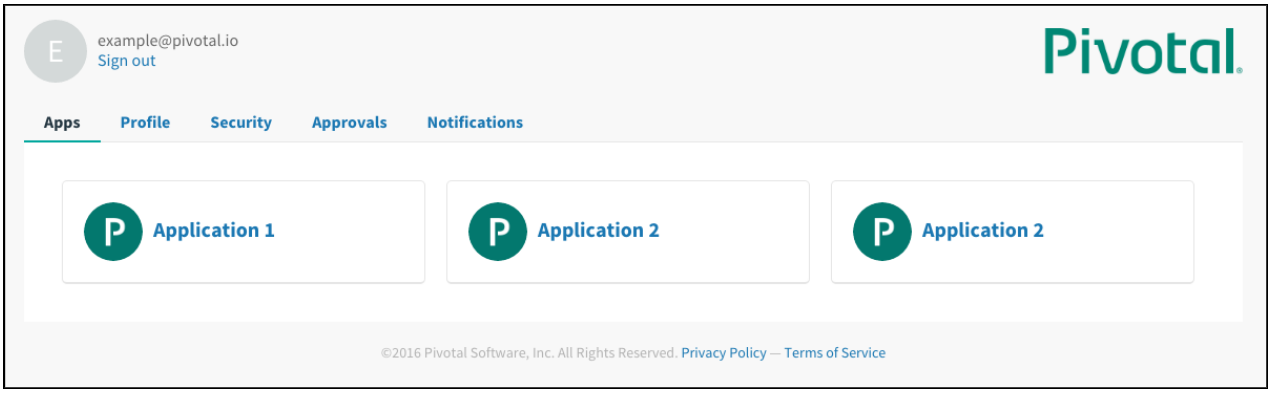
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to PingFederate.



2. Navigate to your application and click it.

3. View the list of applications you have access to.

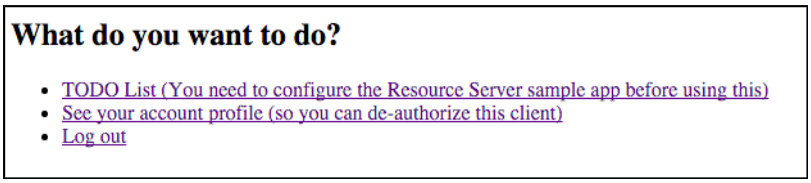


Test Your Single Sign-Off

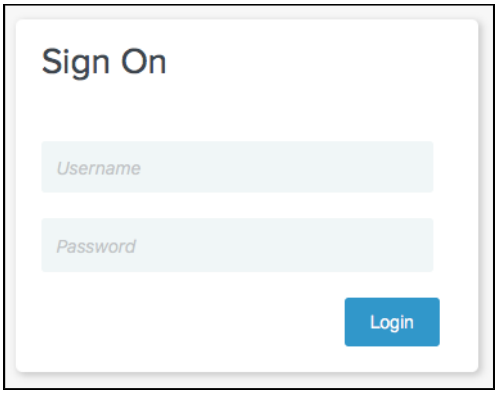
Test single sign-off to ensure that when users log out of the application, they are logged out of PingFederate as well.

1. Sign into the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

2. Under **What do you want to do?**, click **Log out**.



3. Ensure that you are logged out and redirected to the PingFederate login page.

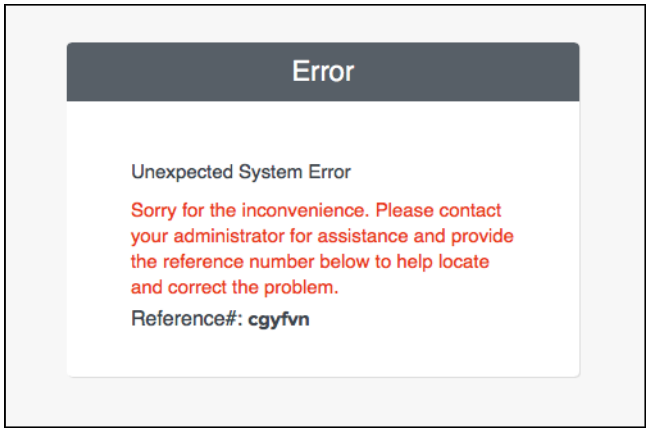


Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between PingFederate and Pivotal Single Sign-On (SSO).

Error

Symptom:

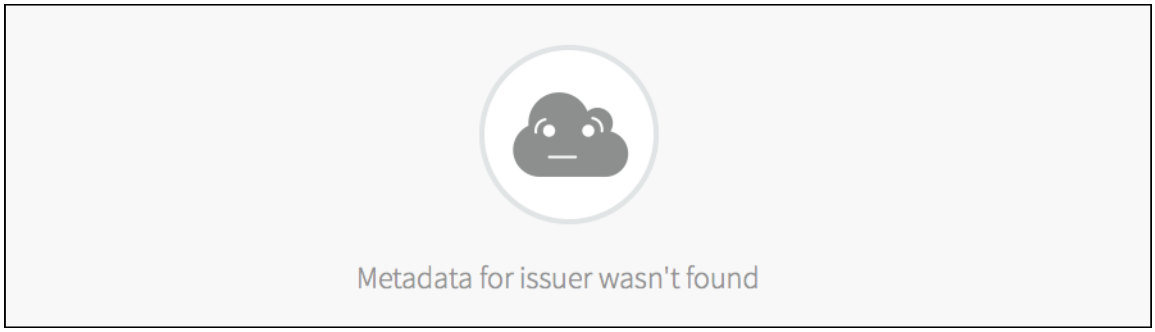


Explanations:

- Connection Status is disabled on PingFederate.
- The service provider Entity ID is misconfigured on PingFederate.
- The identity provider Single Sign-On URL is misconfigured in the SSO plan settings.

Metadata Not Found

Symptom:



Explanation:

- The identity provider Entity ID is misconfigured in the SSO plan settings.

PingOne Cloud Integration Guide Overview

PingOne Cloud is an identity-as-a-service solution that delivers secure single sign-on to SaaS, legacy and web applications. This documentation describes how to configure a single sign-on partnership between PingOne Cloud as the Identity Provider (IdP) and the Single Sign-On Service (SSO) for Pivotal Cloud Foundry as the Service Provider (SP).

SSO supports service provider-initiated authentication flow and single logout. It does not support identity provider-initiated authentication flow. All SSO communication takes place over SSL.

Prerequisites

To integrate PingOne Cloud with Pivotal Cloud Foundry (PCF), you need:

Pivotal

- PCF, version 1.7.0 or later.
- Single Sign-On, version 1.1.0 or later.

PingOne Cloud

- PingOne Cloud
- A user with Application Admin privileges.

Note: To configure SAML, you must have the Single Sign-On service broker installed on your PCF deployment. You need to create a plan, grant any plan administrators, and specify any organizations this plan should be the authentication authority for. For help configuring plans, see the Manage Service Plans topic..

PingOne Cloud Integration Guide

Configuring PingOne Cloud with SSO

Complete both steps below to integrate your deployment with PingOne Cloud and SSO.

1. Configure PingOne Cloud as an Identity Provider
2. Configure a Single Sign-On Service Provider

Testing and Troubleshooting

- Testing
- Troubleshooting

Configure PingOne Cloud as an Identity Provider

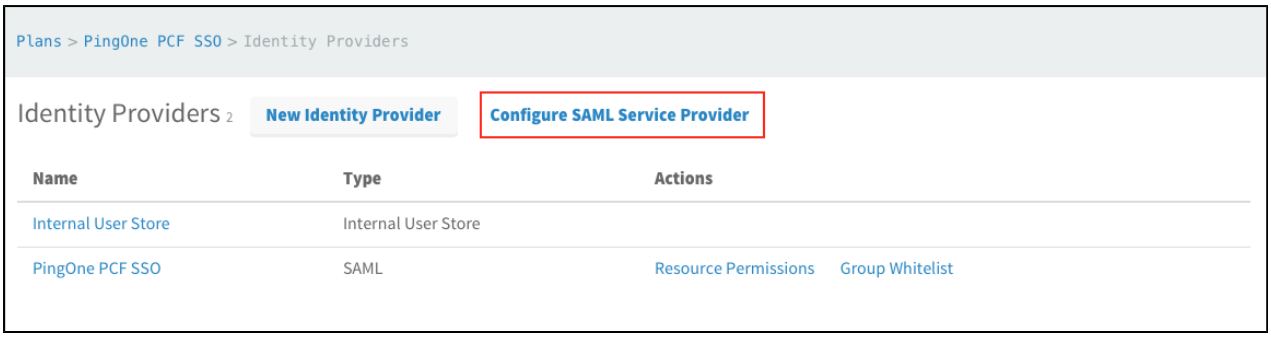
This topic describes how to set up PingOne Cloud as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and PingOne Cloud.

Set up SAML in PCF

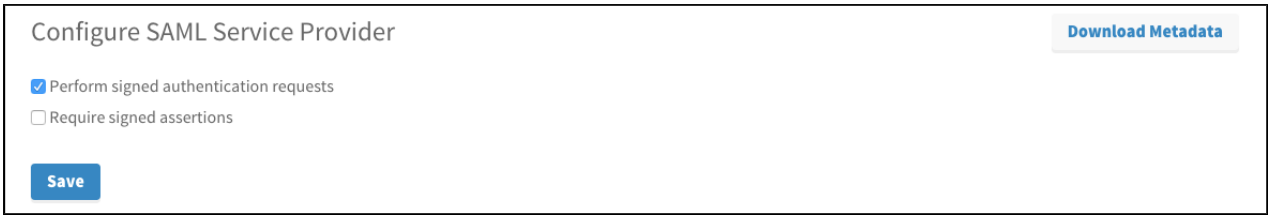
1. Log into the Single Sign-On (SSO) dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **Configure SAML Service Provider**.



4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.

6. Click **Download Metadata** to download the service provider metadata.

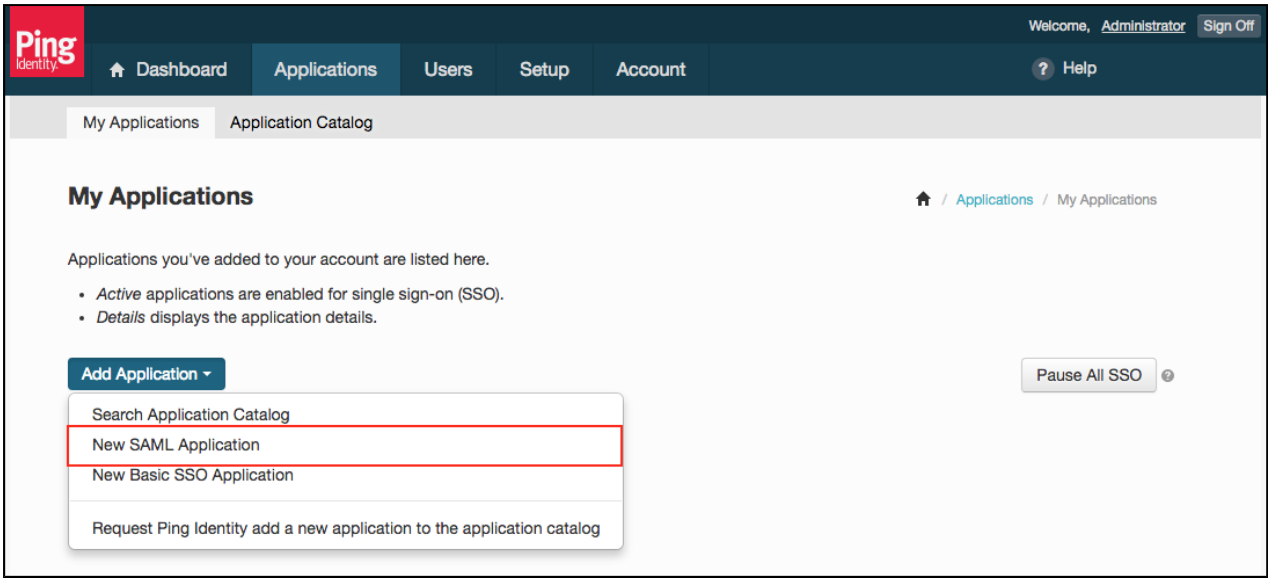
7. Click **Save**.

Set up SAML in PingOne Cloud

1. Sign in as a PingOne Cloud administrator.

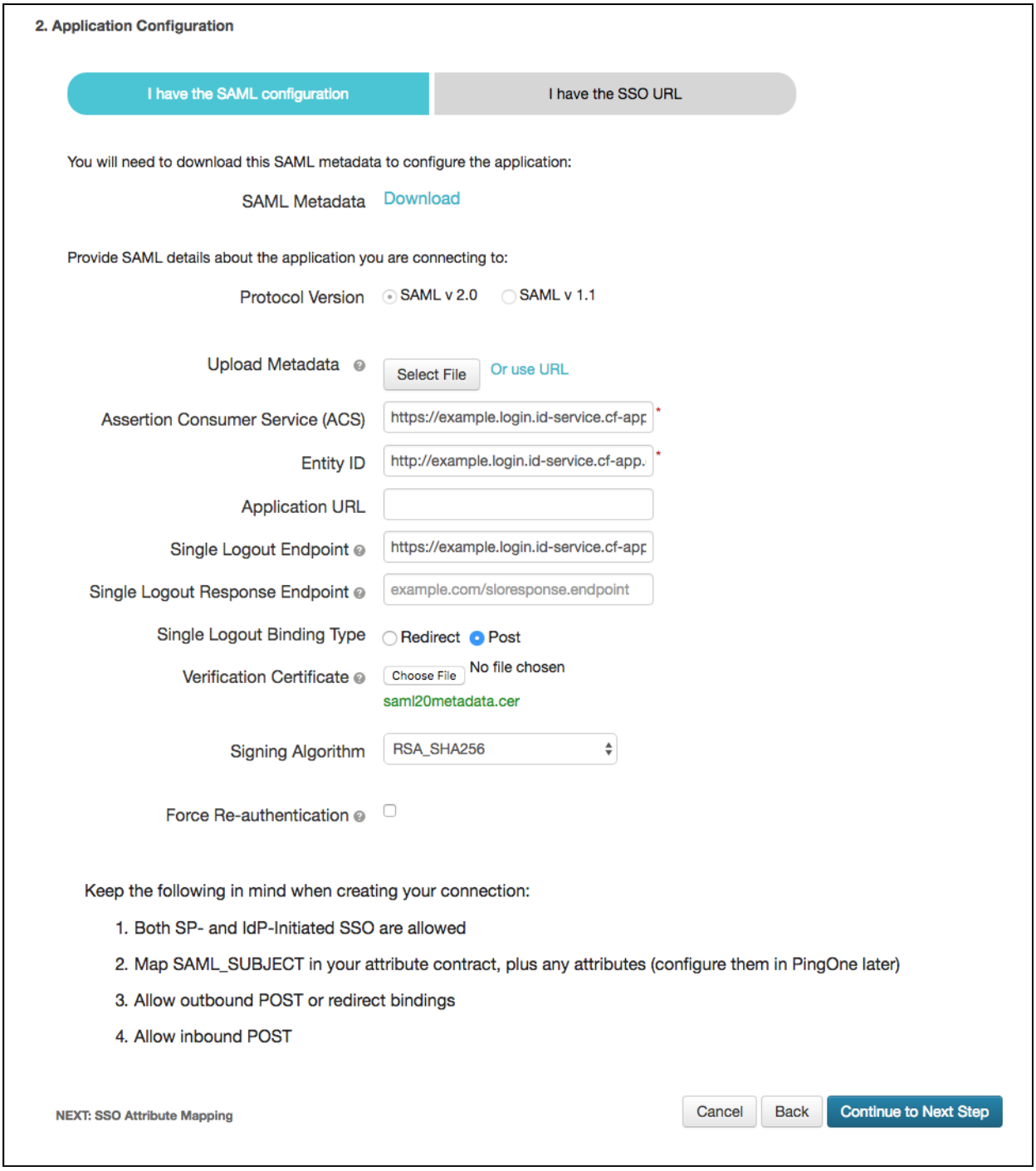
2. Navigate to your application by clicking on the **Applications** tab.

3. Click the **Add Application** button and choose **New SAML Application**.



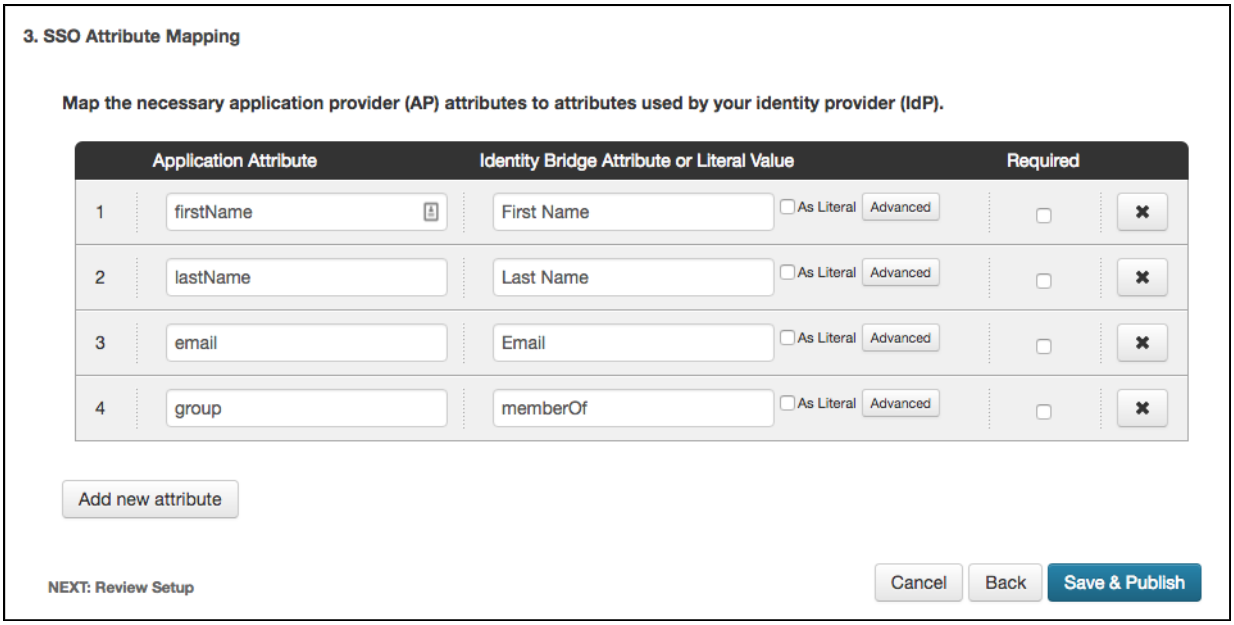
4. Enter the **Application Name**, **Application Description**, **Category** and any **Graphics**.

5. Click the **Continue to Next Step** button to configure SAML.



6. In the **Application Configuration** section, perform the following steps:
    a. Select **I have the SAML configuration**.
    b. For **SAML Metadata**, click **Download** to download the identity provider metadata.
    c. For **Protocal Version**, select **SAML v 2.0**.
    d. For **Upload Metadata**, click **Select File** and select the service provider metadata.
    e. Click the **Continue to Next Step** button.

7. (Optional) Under **SSO Attribute Mapping**, specify any application or group attributes that you want to map to users in the ID token.



8. Click the **Save & Publish** button followed by the **Finish** button.

Configure a Single Sign-On Service Provider

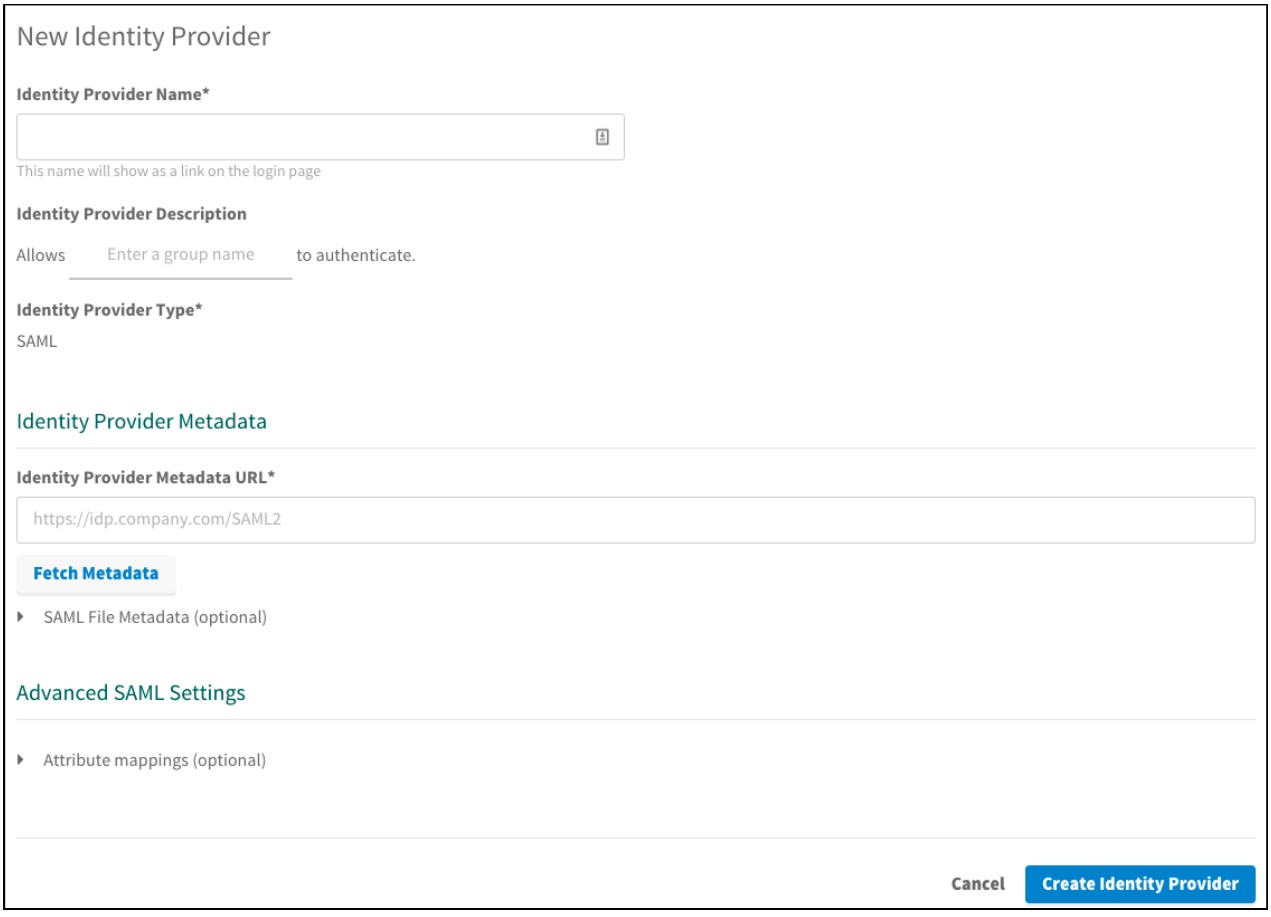
This topic describes how to add an external identity provider to your Pivotal Single Sign-On (SSO) service plan.

Setting up SAML

1. Log into the SSO dashboard at `https://p-identity.YOUR-SYSTEM-DOMAIN` as a Plan Administrator.

2. Select your plan and click **Manage Identity Providers** on the drop-down menu.

3. Click **New Identity Provider** to create a new identity provider.



4. To create a new identity provider, perform the following steps:

    a. Enter an identity provider name into **Identity Provider Name**.
    b. (Optional) Enter a description into **Identity Provider Description**.
    c. Click **SAML File Metadata (optional)** follwed by clicking the **Upload Identity Provider Metadata** button to upload your metadata XML.
    d. (Optional) Under **Advanced SAML Settings**, click **Attribute Mappings** to enter the mappings.

5. Click **Create Identity Provider**.

6. Click **Resource Permissions**.

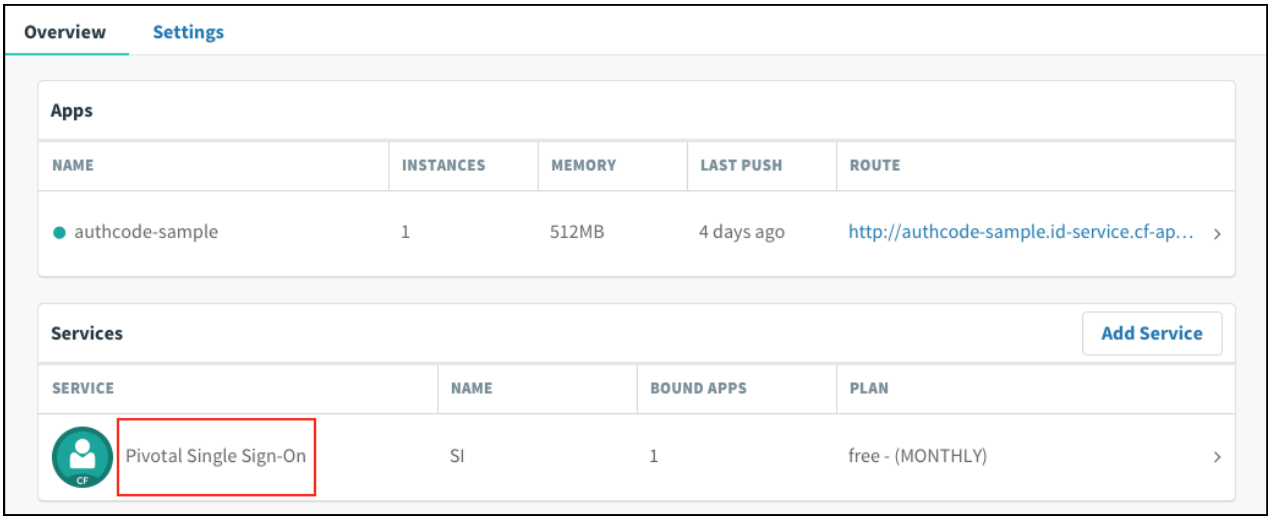
7. Click **New Permissions Mapping** and perform the following steps:
    a. Enter a **Group Name**.
    b. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.
8. Navigate to the identity provider list.
9. Click **Group Whitelist** and enter the group names from the external identity provider that should be propagated in the ID token.

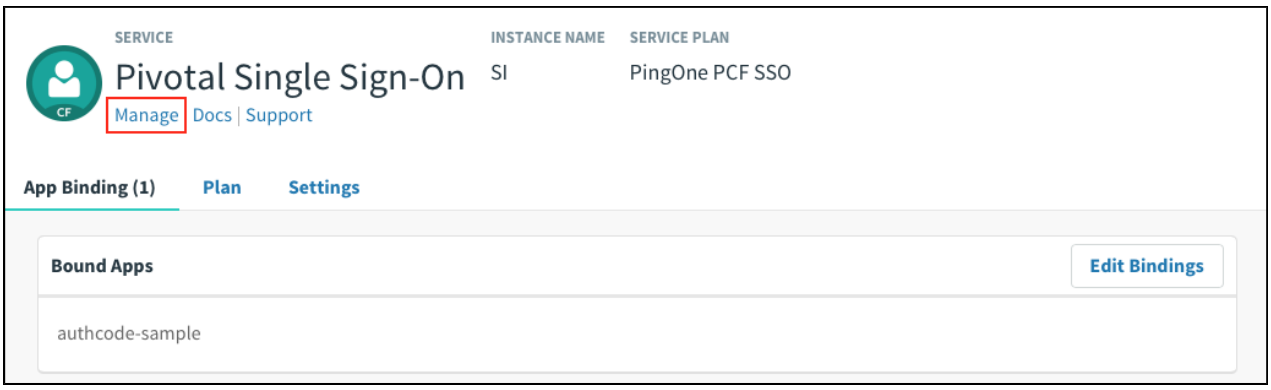
Testing

This topic describes how an administrator can test the connection between SSO and PingOne Cloud. An administrator can test both service provider and identity provider connections.

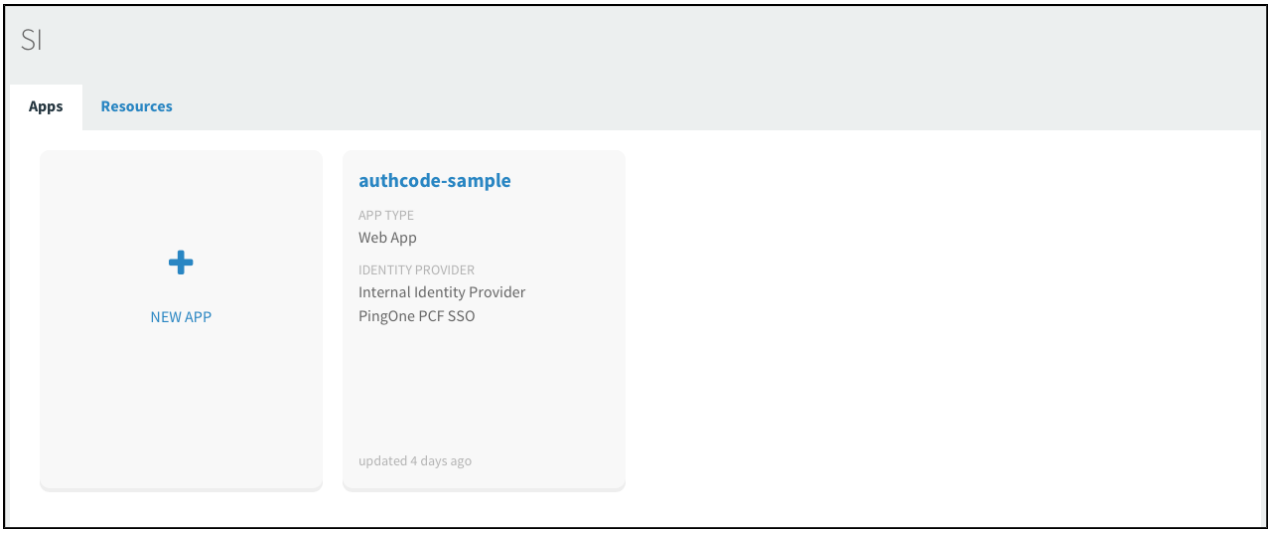
Test Your Service Provider Connection

1. Log in to Apps Manager at https://apps.YOUR-SYSTEM-DOMAIN and navigate to the organization and space where your application is located.
2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Click on the service instance and click **Manage**.

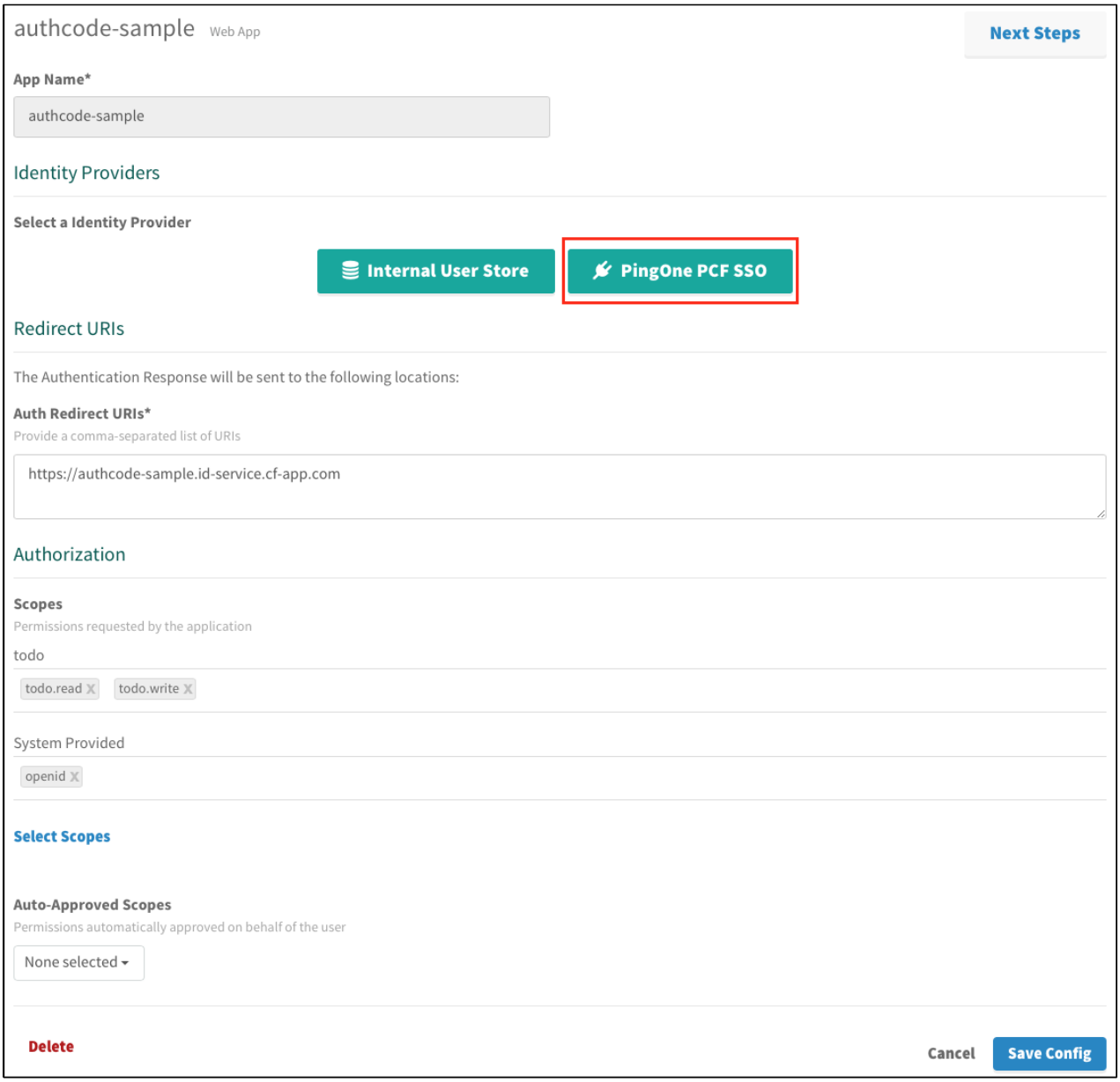




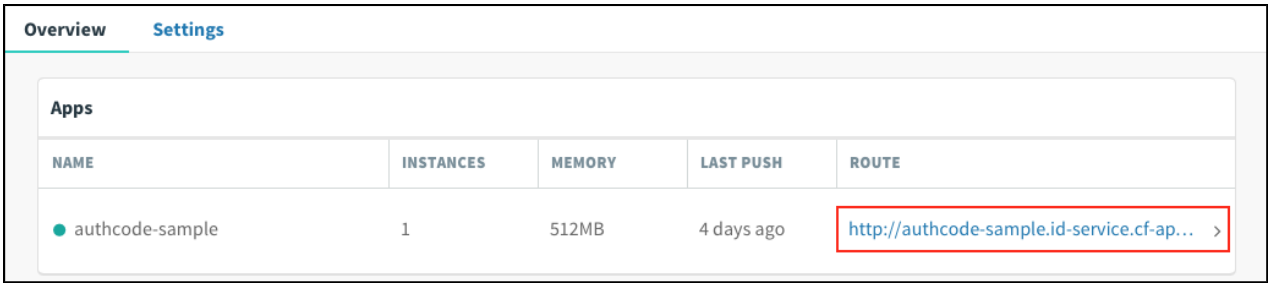
3. Under the **Apps** tab, click your application.



4. Under **Identity Providers**, select the PingOne identity provider.

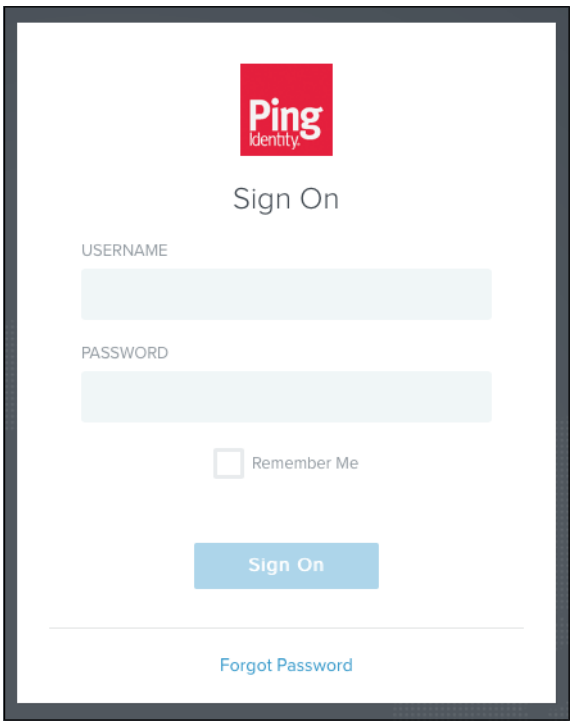


5. Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.

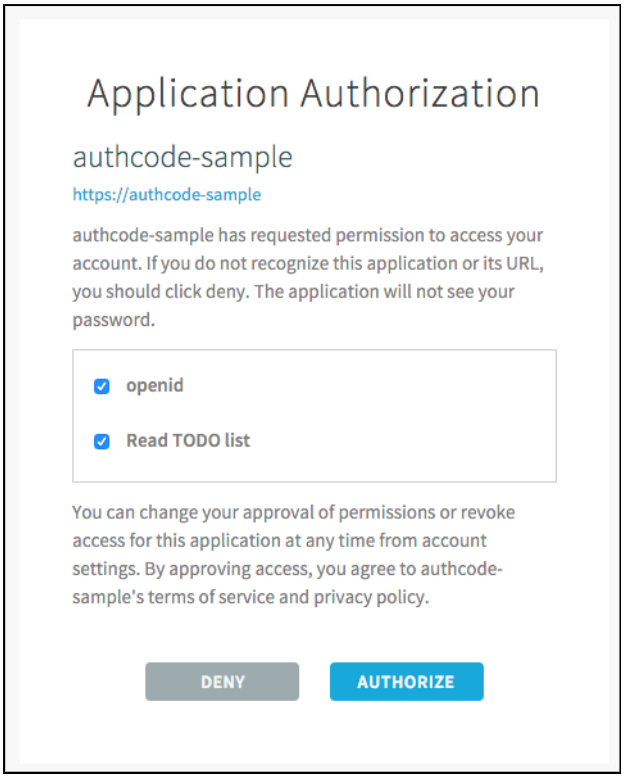


6. Click the link.

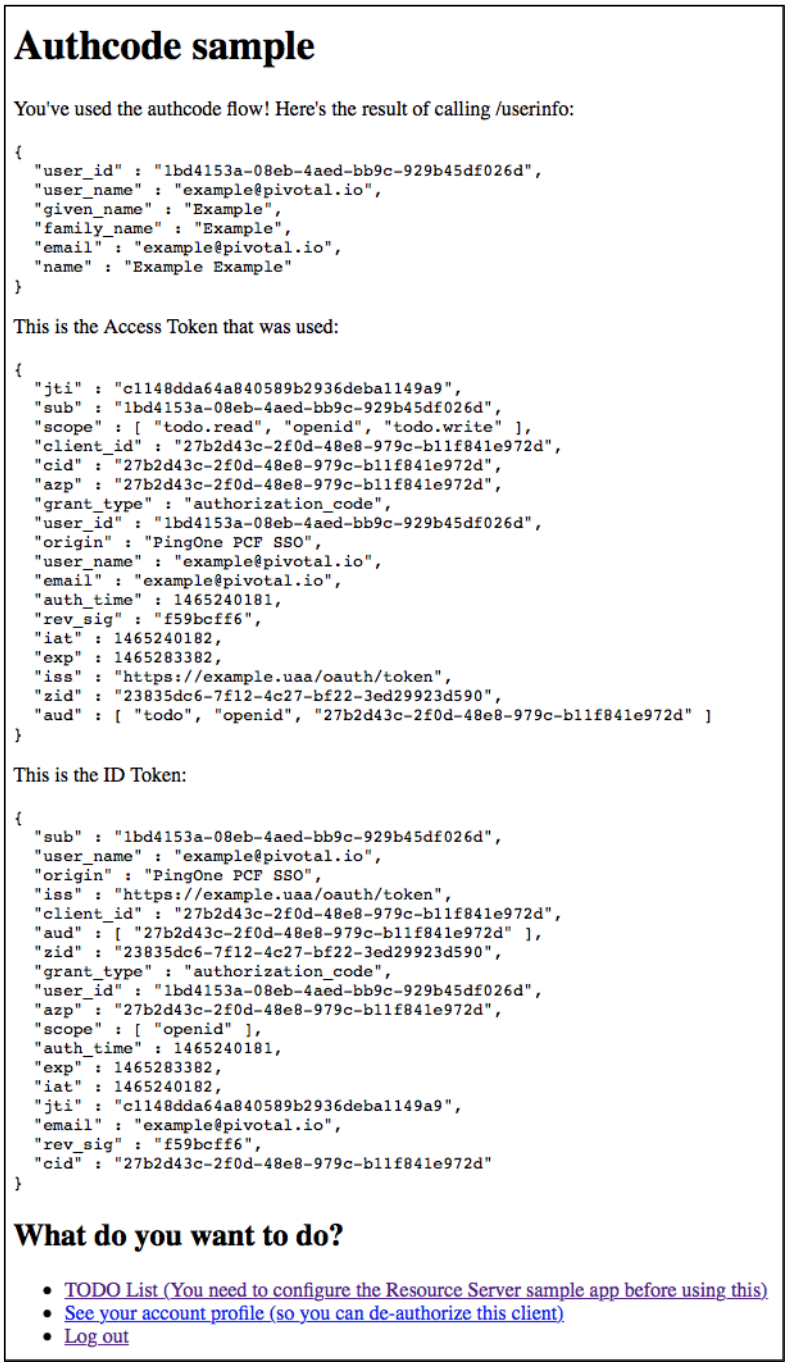
7. On the identity provider sign-in page, enter your credentials and click **Sign On**.



8. The application asks for authorization to the necessary scopes. Click **Authorize**.



9. The access token and ID token displays.



Authcode sample

You've used the authcode flow! Here's the result of calling /userinfo:


```
{
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "user_name" : "example@pivotal.io",
  "given_name" : "Example",
  "family_name" : "Example",
  "email" : "example@pivotal.io",
  "name" : "Example Example"
}
```


This is the Access Token that was used:


```
{
  "jti" : "c1148dda64a840589b2936deba1149a9",
  "sub" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "scope" : [ "todo.read", "openid", "todo.write" ],
  "client_id" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "cid" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "azp" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "grant_type" : "authorization_code",
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "origin" : "PingOne PCF SSO",
  "user_name" : "example@pivotal.io",
  "email" : "example@pivotal.io",
  "auth_time" : 1465240181,
  "rev_sig" : "f59bcff6",
  "iat" : 1465240182,
  "exp" : 1465283382,
  "iss" : "https://example.uaa/oauth/token",
  "zid" : "23835dc6-7f12-4c27-bf22-3ed29923d590",
  "aud" : [ "todo", "openid", "27b2d43c-2f0d-48e8-979c-b11f841e972d" ]
}
```


This is the ID Token:


```
{
  "sub" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "user_name" : "example@pivotal.io",
  "origin" : "PingOne PCF SSO",
  "iss" : "https://example.uaa/oauth/token",
  "client_id" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "aud" : [ "27b2d43c-2f0d-48e8-979c-b11f841e972d" ],
  "zid" : "23835dc6-7f12-4c27-bf22-3ed29923d590",
  "grant_type" : "authorization_code",
  "user_id" : "1bd4153a-08eb-4aed-bb9c-929b45df026d",
  "azp" : "27b2d43c-2f0d-48e8-979c-b11f841e972d",
  "scope" : [ "openid" ],
  "auth_time" : 1465240181,
  "exp" : 1465283382,
  "iat" : 1465240182,
  "jti" : "c1148dda64a840589b2936deba1149a9",
  "email" : "example@pivotal.io",
  "rev_sig" : "f59bcff6",
  "cid" : "27b2d43c-2f0d-48e8-979c-b11f841e972d"
}
```

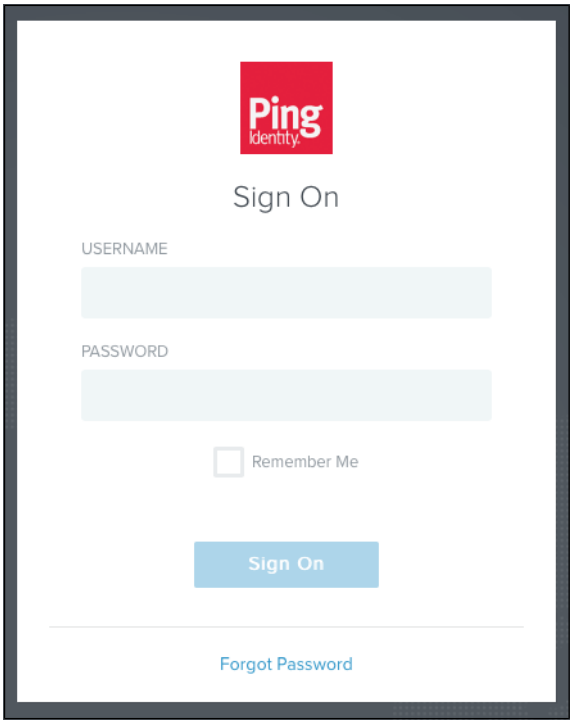

What do you want to do?

- TODO List (You need to configure the Resource Server sample app before using this)
- See your account profile (so you can de-authorize this client)
- Log out

Test Your Identity Provider Connection

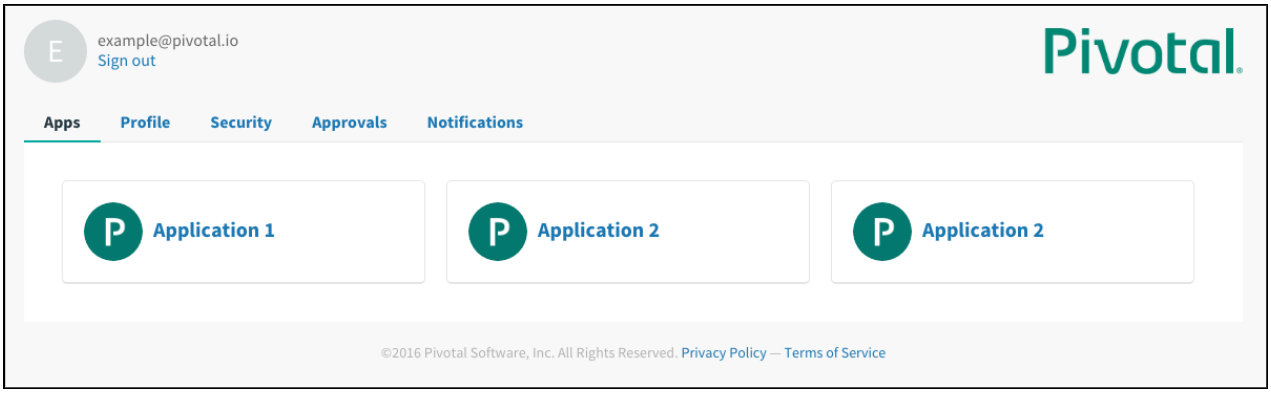
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to PingOne.



2. Navigate to your application and click it.

3. You are redirected to the page that lists applications you have access to.

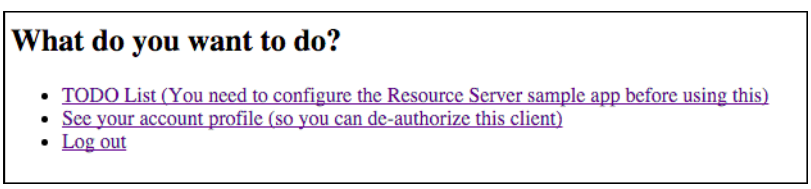


Test Your Single Sign-Off

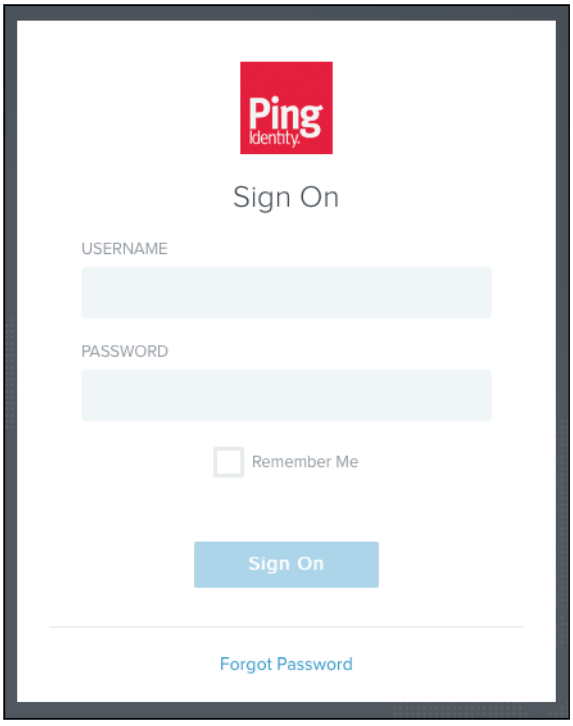
Test single sign-off to ensure that when users log out of the application, they are logged out of PingOne as well.

1. Sign into the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

2. Under “What do you want to do?”, click **Log out**.



3. You are logged out and redirected to the PingOne login page.

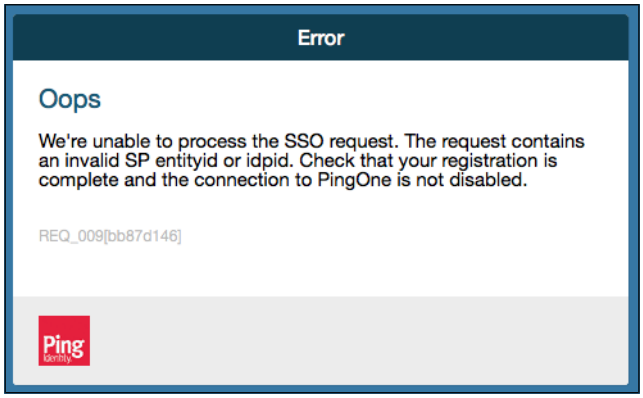


Troubleshooting

This topic describes how to resolve common errors that arise when configuring a single sign-on partnership between PingOne Cloud and Pivotal Single Sign-On (SSO).

Error

Symptom:

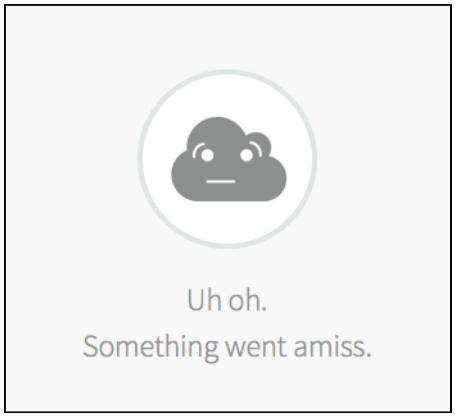


Explanations:

- Single Sign-On is disabled on PingOne.
- The service provider Entity ID is misconfigured on PingOne.
- The identity provider Single Sign-On URL is misconfigured in the SSO plan settings.

Something went amiss

Symptom:



Explanation:

- The service provider Assertion Consumer Service (ACS) is misconfigured on PingOne.

Metadata Not Found

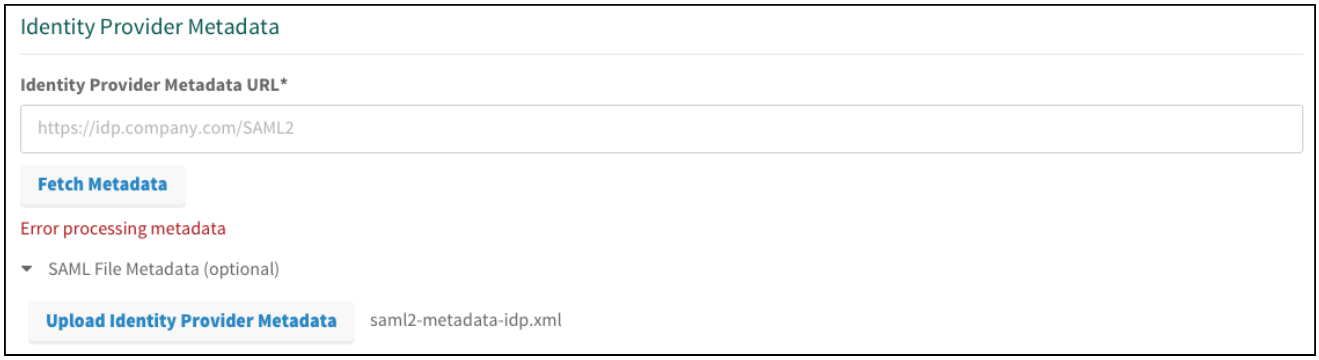
Symptom:

Explanation:

- The identity provider Entity ID is misconfigured in the SSO plan settings.

Missing Name ID

Symptom:



Explanation:

- The identity provider metadata is missing configurations for Name ID. See Configure Identity Provider Metadata.