

Single Sign-On for VMware Tanzu Application Service 1.6

Single Sign-On for VMware Tanzu Application service 1.6

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2023 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

About Single Sign-On	15
Single Sign-On Overview	15
Single Sign-On	15
OAuth 2.0 Authorization	15
Product Snapshot	15
Upgrading to the Latest Version	16
Integration Guides	17
Useful Links	17
Release Notes	17
v1.6.0	18
Features	18
Known Issues	18
Viewing Release Notes for Another Version	18
Getting Started with Single Sign-On	18
Install and Set Up SSO for Applications	19
SSO User Roles	19
Using SSO for Pivotal Cloud Foundry Components	19
Operator Guide	21
Installing Single Sign-On	21
Prerequisites	21
Install SSO via Ops Manager	21
Update SSL and Load Balancer	22
Configure Application Security Groups	22
Using the System Plan	22
System Plan Best Practice	23
Administrators: Configure the System Plan for an Org	23
Developers: Create a System Plan Instance for your App	24
Developers: Revoke System Plan Access for an External App	25
Managing Service Plans	26

Create or Edit Service Plans	26
Delete Service Plans	28
Configure a Token Policy	28
Managing Service Plans with UAAC API	28
Create a UAA Identity Zone Admin Client	29
Create an Admin Client	29
Create a UAA Identity Zone Admin Client	29
Update UAA Identity Zone Configurations with the API	31
Modify Branding	33
Add Default Groups for Users	33
Rotate JSON Web Token (JWT) Signing Keys	34
Configuring Identity Providers	35
Configure Internal User Store	35
Add Internal Users From the Command Line	36
Define Password Policy for the Internal User Store	37
Configure Service Provider SAML Settings	38
Add an External Identity Provider	38
Add a SAML Provider	38
Add an OIDC Provider	39
Add an LDAP Identity Provider	40
Delete an External Identity Provider	41
Configure Group Allowlist for an External Identity Provider	42
Enabling Identity Provider Discovery	42
What it Does	43
Example	43
Enable IdP Discovery	43
Managing Users	44
Manage Users in an Internal User Store	45
Manage Users from an External Identity Provider	46
Manage Users with the UAA CLI	48
About Performance	49
Backing Up and Restoring	50
Developer Guide	51

Determining SSO Application Type	51
OAuth 2.0 Grant Types	51
Authorization Code Grant Type	52
Authorization Code Grant Type Roles	52
Authorization Code Grant Type Flow	52
Client Credentials Grant Type	53
Client Credentials Grant Type Roles	53
Client Credentials Flow	53
Resource Owner Password Grant Type	54
Resource Owner Password Grant Type Roles	54
Resource Owner Password Flow	55
Implicit Grant Type	55
Implicit Grant Type Roles	55
Implicit Flow	56
Common App Architecture Patterns	57
External Apps Calling into PCF APIs	57
UAA Authorization Code Grant — Browser	58
SAML Bearer Token Exchange — Back End	58
Example SAML Assertion	59
JWT Bearer Token Exchange	61
Example JWT Token Content	61
Handling the JWT Token Exchange Using a Gateway	62
Set Up for SAML or JWT Bearer Token Exchange	62
Managing Service Instances	63
Create Service Instances	64
Delete Service Instances	64
Configuring Apps	65
Set Up PCF Apps to Use SSO	65
Configure SSO Properties	65
Remove SSO Configuration Properties	67
Manually Configure Apps for SSO	67
Bootstrap SSO Configuration	68
Bind a PCF App	68
Manage App Configurations via SSO Dashboard	69
Register an External App	70
Integrate SSO with an App	72

Java Apps	72
Non-Java Apps	72
Create Admin Client	72
Delete App that Uses SSO	73
Delete a PCF App	73
Delete an External App	73
Managing Resources	74
Create or Edit Resources	74
Delete Resources	75
Create or Edit Resource Permissions	75
Delete Resource Permissions	76
About Space Protection for Resources	76
Integration Guides	77
Active Directory Federation Services Integration Guide	77
Active Directory Federation Services Integration Guide Overview	77
Prerequisites	77
Active Directory Federation Services Integration Guide	77
Configuring AD FS with SSO	77
Testing and Troubleshooting	78
Configuring Active Directory Federation Services as an Identity Provider	78
Set Up SAML in PCF	78
Set Up SAML in Active Directory Federation Services	79
Setting Up Groups in SAML from ADFS	88
Create Custom Value “groups”	90
Configuring a Single Sign-On Service Provider	93
Download Identity Provider Metadata	93
Setting up SAML	94
Create Attribute Mappings for SAML Groups	95
Testing	95
Test Your Service Provider Connection	96
Test Your Identity Provider Connection	100
Test Your Single Sign-Off	101
Troubleshooting	102
Event Viewer	102

Azure Active Directory SAML Integration Guide	103
Azure Active Directory SAML Integration Guide Overview	103
Prerequisites	104
Azure AD Integration Guide	104
Configuring Azure AD with SSO	104
Testing and Troubleshooting	104
Configuring Azure Active Directory as a SAML Identity Provider	104
Step 1: Set up SAML in PCF	104
Step 2: Set up SAML in Azure Active Directory (AD)	105
Step 3: Set up Claims Mapping	108
Configuring a Single Sign-On Service Provider	110
Step 1: Setting up SAML	111
Step 2: Configure Group Permissions	112
Testing	112
Test Your Configurations in Azure AD	113
Test Your Service Provider Connection	114
Test Your Identity Provider Connection	118
Test Your Single Sign-Off	119
Troubleshooting	120
Failed Login	120
Symptom:	120
Solutions:	121
App ID Not Found	121
Symptom:	121
Explanations:	121
Reply URL Does Not Match	121
Symptom:	121
Explanation:	122
Missing Name ID	122
Symptom:	122
Explanation:	122
Azure Active Directory OIDC Integration Guide	122
Azure Active Directory OIDC Integration Guide Overview	122

Prerequisites	123
Azure AD Integration Guide	123
Configuring Azure AD with SSO	123
Testing and Troubleshooting	123
Configuring Azure Active Directory as an OIDC Identity Provider	123
Configuring an OIDC Service Provider in SSO	129
Testing Your Single Sign-On Connection	131
Troubleshooting	135
Bad Request	135
Symptom:	135
Cannot determine username from credentials supplied	135
Symptom:	135
Explanation:	136
Azure Error for Reply Address	136
Symptom:	136
Explanation:	137
Login Page Cannot Be Found (404 Error)	137
Symptom:	138
Explanation:	138
Error authenticating against external identity provider: 404 Not Found	138
Symptom:	138
Explanation:	138
Error authenticating against external identity provider: Invalid issuer for token did not match expected	138
Symptom:	138
Explanation:	139
Request Method 'POST' not supported (405 Error)	139
Symptom:	139
Explanation:	139
Error authenticating against external identity provider: Some parties were not in the token audience	139
Symptom:	139
Explanation:	139
CA Single Sign-On Integration Guide	140
CA Single Sign-On Integration Guide Overview	140
Prerequisites	140

CA Single Sign-On Integration Guide	140
Configuring CA Single Sign-On with SSO	140
Testing and Troubleshooting	140
Configuring CA Single Sign-On as an Identity Provider	141
Set up SAML in PCF	141
Set up SAML in CA Single Sign-On	142
Configuring a Single Sign-On Service Provider	146
Setting up SAML	146
Testing	147
Test Your Service Provider Connection	148
Test Your Identity Provider Connection	152
Test Your Single Sign-Off	153
Troubleshooting	154
CA Single Sign-On Partnership is Inactive	154
Symptom:	154
Explanations:	154
Service Provider Entity ID Misconfigured	154
Symptom:	154
Explanation:	154
Incoming SAML message is invalid	155
Symptom:	155
Explanation:	155
Assertion Consumer Service URL Misconfigured	155
Symptom:	155
Explanation:	155
Audience Field Misconfigured	155
Symptom:	155
Explanation:	155
Expired Certificate	155
Symptom:	155
Explanation:	156
Identity Provider SSO URL Misconfigured	156
Symptom:	156
Explanation:	156
Google Cloud Platform OIDC Integration Guide	156

Google Cloud Platform OIDC Integration Guide Overview	156
Prerequisites	156
Integrate Google Cloud Platform OIDC for SSO	157
Test and Troubleshoot	157
Configuring GCP as an OIDC Identity Provider	157
Generate GCP Client Credentials	157
Set Up OIDC Identity Provider in SSO	159
Testing	161
Test Your Single Sign-On Connection	162
Troubleshooting	164
No Link for OIDC	165
Symptom:	165
Explanation:	165
No OAuth Client Found	165
Symptom:	165
Explanation:	166
Unauthorized	166
Symptom:	166
Explanation:	166
Redirect URI Mismatch	166
Symptom:	166
Explanation:	167
Empty Username	167
Symptom:	167
Explanation:	167
Unable to map claim to a username	167
Symptom:	167
Explanation:	168
Okta Integration Guide	168
Okta Integration Guide Overview	168
Prerequisites	168
Okta Integration Guide	169
Configuring Okta with SSO	169
Testing and Troubleshooting	169
Configuring Okta as an Identity Provider	169

Set up SAML in PCF	169
Set Up SAML in Okta	170
Configuring a Single Sign-On Service Provider	173
Setting up SAML	174
Testing	175
Test Your Service Provider Connection	175
Test Your Identity Provider Connection	180
Test Your Single Sign-Off	181
Troubleshooting	182
Page Not Found	182
Symptom:	182
Explanations:	183
No Valid Assertion	183
Symptom:	183
Explanations:	183
Webpage Not Available	183
Symptom:	183
Explanation:	184
Metadata Not Found	184
Symptom:	184
Explanation:	184
PingFederate Integration Guide	184
PingFederate Integration Guide Overview	184
Prerequisites	185
PingFederate Integration Guide	185
Configuring PingFederate with SSO	185
Testing and Troubleshooting	185
Configuring PingFederate as an Identity Provider	185
Set up SAML in PCF	186
Set up SAML in PingFederate	186
Configure the Connection	186
Configure Browser SSO	188
Assertion Creation	188
Protocol Settings	190
Configure Credentials	190

Configuring a Single Sign-On Service Provider	191
Setting up SAML	191
Testing	193
Test Your Service Provider Connection	193
Test Your Identity Provider Connection	197
Test Your Single Sign-Off	198
Troubleshooting	199
Error	199
Symptom:	199
Explanations:	200
Metadata Not Found	200
Symptom:	200
Explanation:	200
PingOne Cloud Integration Guide	200
PingOne Cloud Integration Guide Overview	200
Prerequisites	201
PingOne Cloud Integration Guide	201
Configuring PingOne Cloud with SSO	201
Testing and Troubleshooting	201
Configuring PingOne Cloud as an Identity Provider	201
Set up SAML in PCF	202
Set up SAML in PingOne Cloud	202
Configuring a Single Sign-On Service Provider	205
Setting up SAML	205
Testing	206
Test Your Service Provider Connection	207
Test Your Identity Provider Connection	211
Test Your Single Sign-Off	212
Troubleshooting	213
Error	214
Symptom:	214
Explanations:	214

Something went amiss	214
Symptom:	214
Explanation:	215
Metadata Not Found	215
Symptom:	215
Explanation:	215
Missing Name ID	215
Symptom:	215
Explanation:	216
Plan-to-Plan OIDC Integration Guide	216
Plan-to-Plan OIDC Integration Guide	216
Prerequisites	216
Integrating a Plan-to-Plan OIDC for SSO	217
Testing the OIDC Connection	217
Troubleshooting	217
Configuring Plan-to-Plan OIDC Integration	217
Setting Up Relying Party Configurations in the Identity Provider Plan	217
Prerequisites	217
Setting Up the OIDC Identity Provider Configuration in the Relying Party Plan	219
Finalizing Configuration	221
Testing OIDC Integrations	221
Testing Your SSO Connection	221
Troubleshooting Plan-to-Plan OIDC Integration	223
No link for OIDC, or the Service Provider Login page is blank	223
Cause	224
Authorization Request Error	224
Cause	225
401 Unauthorized	225
Cause	225
405 Method Not Allowed	225
Cause	225
Cannot determine username with given credentials	225
Cause	226
Invalid redirect	226
Cause	226

About Single Sign-On

Single Sign-On Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). 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 [Configuring 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	v1.6.0
Release date	April 19, 2018
Compatible Ops Manager version(s)	v2.0 and v2.1
Compatible Pivotal Application Service (PAS) version(s)	v2.0 and v2.1
IaaS support	AWS, GCP, OpenStack, Azure, 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 v2.0 to PCF v2.1, upgrade PCF so that SSO v1.5.3 is running on PCF v2.1, and then upgrade SSO v1.5.3 to SSO v1.6 as soon as possible.



Breaking Change: You can only upgrade to SSO v1.6 on PCF v2.0.
For information about upgrading to PCF v2.0, see [Upgrading Pivotal Cloud Foundry](#).

Elastic Runtime* or PAS Version	Supported Upgrades from SSO Versions	
	From	To
1.10.x	1.2.0 – 1.2.4	1.3.6
	1.3.0-1.3.5	
1.11.x	1.3.0-1.3.6	1.4.6
	1.4.1-1.4.5	
1.12.x	1.4.1-1.4.6	1.5.3
	1.5.0-1.5.2	
2.0.x	1.5.3	1.5.x
		1.6.0
2.1.x	1.6.0	1.6.x

* As of PCF v2.0, *Elastic Runtime* is renamed *Pivotal Application Service (PAS)*.



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

- The SSO v1.4.x tiles are compatible with PCF v1.11.x and v1.12.x.
- The SSO v1.5.x tiles are compatible with PCF v1.12.x.
- The SSO tiles v1.5.3 and later are compatible with PCF v1.2.x, v2.0.x, and v2.1.x.

- The SSO v1.6.x tiles are compatible with PCF v2.0.x and v2.1.x.

Integration Guides

Use these guides to help you plan and implement your integration with the Single Sign-On service for PCF.

- [Active Directory Federation Services \(AD FS\) Integration Guide](#)
- [Azure Active Directory SAML Integration Guide](#)
- [Azure Active Directory OIDC Integration Guide](#)
- [CA Single Sign-On Integration Guide](#)
- [Google Cloud Platform OpenID Connect Integration Guide](#)
- [Okta Integration Guide](#)
- [PingFederate Integration Guide](#)
- [PingOne Cloud Integration Guide](#)
- [Plan-to-Plan OIDC Integration Guide](#)

Useful Links

- [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](#)
- [Configuring Applications](#)
- [Manage Resources](#)

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Release Notes



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

These are release notes for the [Single Sign-On](#) service for Pivotal Cloud Foundry (PCF).

v1.6.0

Release Date: April 19, 2018

Features

New features included in this release:

- **Expanded API Functionality:** Operators can now use the API to configure plan features that are not yet exposed through the SSO dashboard. The SSO service honors values configured through the API so that these values are retained when using the SSO dashboard. For more information, see the following sections of [Manage Service Plans with UAA API](#):

To learn more about...	See...
Updating the branding of your login page	Enable Branding
Adding default groups for users	Add Default Groups for Users
Rotating JWT signing keys	Rotate JWT Signing Keys

- **Plan Visibility for All Orgs:** Operators can now enable an SSO service plan to be visible to all orgs. Operators may want to use this feature to provide a test plan to all developers as a default. Operators can enable or disable this mode for any new or existing service plan.
- **BOSH Backup and Restore Compatibility:** SSO now works with BOSH Backup and Restore (BBR). For more information, see [Backing Up and Restoring](#).



WARNING: BBR will introduce about a minute of downtime for application login during a backup operation on PCF 2.0. Starting with PCF v2.1 and later, there is no application downtime when performing a BBR backup.

- **Availability of Performance Data:** Performance benchmarks, metrics, and scaling indicators are now available. For more information, see [About Performance](#).

Known Issues

No new issues known for this release.

Viewing 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Getting Started with Single Sign-On



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates,

upgrade to a supported version.

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](#). Single Sign-On is a multi-tenant service and a service plan corresponds to a tenant. This allows an enterprise to separate 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 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 Pivotal Cloud Foundry apps as well as externally hosted apps. 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 adds the endpoints to an allowlist 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Operator Guide

Installing Single Sign-On



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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



Breaking Change: You can only upgrade to SSO v1.6 on PCF v2.0. For information about upgrading to PCF v2.0, see [Upgrading Pivotal Cloud Foundry](#).

Prerequisites

- [Ops Manager](#) v2.0, v2.1, or later and [Pivotal Application Service](#) v2.0 or v2.1.
- SSL Certificates.
- Application Security Groups (ASGs).

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 ASG 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 how to set up ASGs, see [Application Security Groups](#).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Using the System Plan



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 Pivotal Application Service 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 Pivotal Application Service 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).

P

Single Sign-On

admin ▾

Plans > System

System

Plan Name*

System

Description*

This plan is reserved for app developer single sign-on.

This will appear as a plan feature in the Apps Manager Marketplace

Auth Domain* <https://login.sys.banana.gcp.releg.cf-app.com>

Organizations

MY-ORG ⓘ

MY-ORG

sree-org ✕

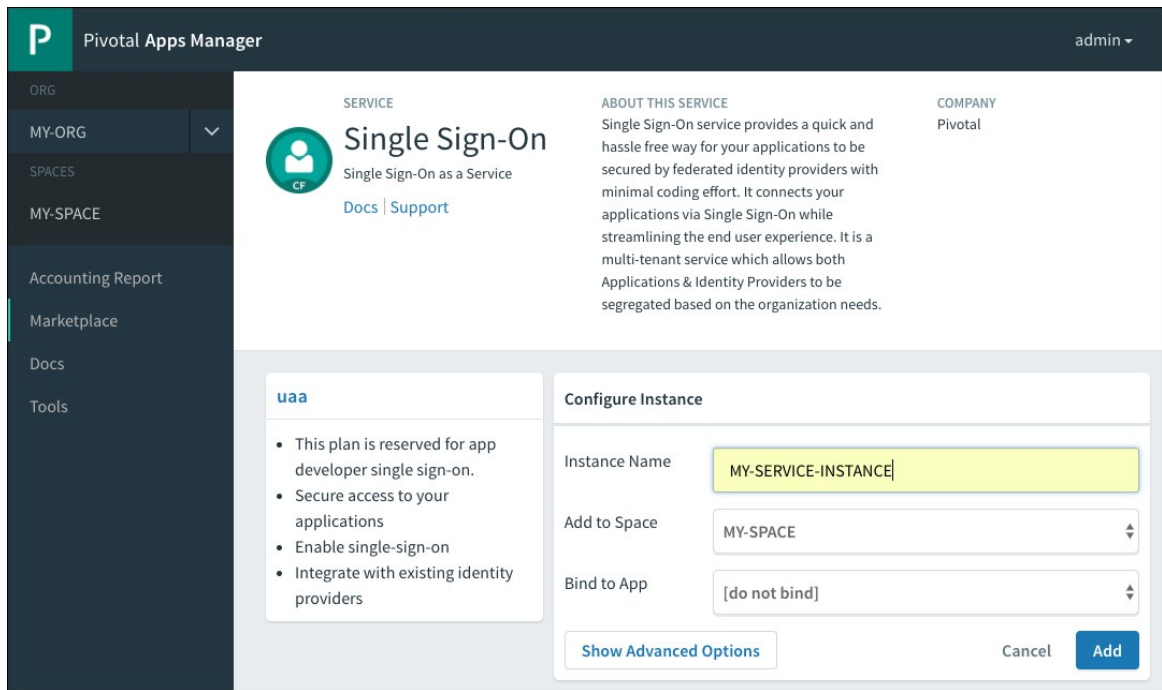
Cancel

Save Plan

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 you have a PCF app, bind the application with the service instance you created. For more information, see [Bind a PCF App](#).
3. If your app is a pipeline or a script that runs external to PCF but calls PCF APIs, do the following:
 1. Follow the instructions to [Register an Externally Hosted App Using the SSO Developer Dashboard](#) 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.
 2. To give your pipeline or script access to your resources without your presence, embed a refresh token instead of hardcoding your credentials:
 1. Run `uaac token sso get`.
 2. 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.
 3. Paste the authentication URL into a browser, and log in using your UAA Admin Credentials.
 4. Copy the **Temporary Authentication Code** from the browser into the UAAC to finish the authentication.
 5. Run `uaac context`.
 6. 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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Managing Service Plans



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Single Sign-On (SSO) 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 isolate 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

You may also want to configure an SSO Service Plan as an OpenID Connect (OIDC) identity provider. For more information, see [Plan-to-Plan OIDC Integration Guide](#).

Create or Edit Service Plans

Administrators can create new SSO service plans at any time from the SSO dashboard. 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 the Pivotal Application Service (PAS) tile in Ops Manager in the **Credentials** tab.
2. Click **New Plan** on the SSO dashboard to create a new SSO service plan.

Single Sign-On admin

[< Back](#)

Create Plan

[Cancel](#) [Create Plan](#)

Plan Name*

SSO Plan

Description*

This is a Single Sign-On Service Plan description.

This will appear as a plan feature in the Apps Manager Marketplace

Auth Domain* https://sso-auth-domain.login.domain.cf-app.com

sso-auth-domain

Instance Name*

SSO Login for Apps

This will appear as a title on the Sign In page

Plan Administrators

Add Users

demo-admin

Organizations

Add Orgs

demo

☐ Enable for all Orgs

[Cancel](#) [Create 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 **Organizations**, select specific organizations in your PCF deployment that can access your Single Sign-On service plan, or select **Enable for all Orgs**.
 - ◆ If you select **Enable for all Orgs** the plan is available for use and displayed in the Services Marketplace for all developers in any organization. This is only recommended for test plans to allow developers to experiment with the SSO service.
 - ◆ If you do not select any organizations, the plan is not available for use and it is not

displayed in the Services Marketplace.

9. Click **Create Plan**. Your new plan appears in the Services Marketplace in the organizations you 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



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.

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 Application Service 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.

Configure a Token Policy

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

- **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.

To configure the token policy, do the following:

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 Application Service (PAS) tile in Ops Manager in the **Credentials** tab.
2. Select the name of the plan you want to configure a token policy for, and click **Configure** from 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**.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Managing Service Plans with UAAC API



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how Pivotal Cloud Foundry (PCF) admins or plan admins can manage Single Sign-On (SSO) service plan configurations using the SSO Operator Dashboard or API using the User Account and Authentication Command Line Interface (UAAC).

The SSO service for PCF manages configurations within the UAA and the Cloud Controller (CC) components of the Pivotal Application Service (PAS). Each SSO service plan ties together a CC plan and a UAA identity zone.

Beginning with SSO v1.6, you can use the UAAC to manage UAA identity zones configured as part of SSO service plans.

Create a UAA Identity Zone Admin Client

To use the UAAC with your SSO service plan, you need an identity zone admin client. To create the identity zone admin client, you need to create a UAA admin client that corresponds to your SSO service plan.

Create an Admin Client

To create an UAA admin client, do the following:

1. Follow the procedure in [Create an Admin Client](#).
2. Using the instructions above, give this client the `clients.admin` scope.
3. Record the **App ID** and **App Secret**. You need these for the procedure below.

Create a UAA Identity Zone Admin Client

To create a UAA identity zone admin client:

1. Install the UAAC as follows:

```
gem install cf-uaac
```

For information about the UAAC, see [the UAAC Github Repository](#).

2. Use the UAAC to target your service plan:

```
uaac target MY-AUTH-DOMAIN.login.example.com
```

Where `MY-AUTH-DOMAIN` is the **Auth Domain** you entered when you created the [Service Plan](#).

3. Run the command below to authenticate and obtain an access token for the admin client for your service plan. UAAC stores the token in `~/.uaac.yml`.

```
uaac token client get MY-APP-ID -s MY-APP-SECRET
```

Where:

- ✦ **MY-APP-ID** is your admin app ID.
- ✦ **MY-APP-SECRET** is your app secret.

Use the **App ID** and **App Secret** provided when you created the admin client in the procedure above.

4. Run the following command to create an identity zone admin client.

```
uaac client add ZONE-ADMIN-CLIENT-ID --authorized_grant_types client_credentials --authorities uaa.admin
```

Where **ZONE-ADMIN-CLIENT-ID** is an ID you want to use to identify this zone admin client.

When prompted for a **New client secret**, provide a client secret for this identity zone admin client. Ensure you use a secure value for your client secret.

For example:

```
$ uaac client add ExampleZoneAdminClientID --authorized_grant_types client_credentials --authorities uaa.admin
New client secret: *****
Verify new client secret: *****
```

Record the values you provide for **ZONE-ADMIN-CLIENT-ID** and **New client secret**.

You can delete the original admin client created through the SSO UI after you create the identity zone client.

5. Run the following command to authenticate and obtain an access token for the identity zone admin client for your service plan.

```
uaac token client get ZONE-ADMIN-CLIENT-ID
```

Where **ZONE-ADMIN-CLIENT-ID** is zone admin client ID you provided in the previous step.

When prompted for a **Client secret**, use the client secret you provided in the previous step.

For example:

```
$ uaac token client get ExampleZoneAdminClientID
Client secret: *****
```

6. Use the following command to display your client context and verify that you have **uaa.admin** under the scope section.

```
uaac context
```

For example:

```
$ uaac context
[1]*[ExampleZoneAdminClientID]
  client_id: ExampleZoneAdminClientID
  access_token: asdioqwuelk12312.e21e
  token_type: bearer
  expires_in: 43200
  scope: uaa.admin
  jti: 123908dk11-23298
```

You can now do operator level API configurations for the SSO service plan. You do not have permissions for any other SSO service plan.

Update UAA Identity Zone Configurations with the API

This section shows how to use the UAAC to update UAA identity zone configurations, using a **PUT** command.



WARNING: This flow is for advanced users only. You must always run the **PUT** command with the latest data by doing a **GET** before a **PUT** command. You must also provide all configuration values, otherwise, data might be lost.

For general information about UAA API, see the [CF UAA API documentation page](#).

To make UAA identity zone API calls, do the following:

1. Create an identity zone admin client following [Create a UAA Identity Zone Admin Client](#) above.
2. Run the following command, directing the output to a text file:

```
uaac curl -k /identity-zones/ZONE-ADMIN-CLIENT-ID > JSON-BLOB.txt
```

Where:

- ♦ **ZONE-ADMIN-CLIENT-ID** is your identity zone admin client ID created in [Create a UAA Identity Zone Admin Client](#) above.
 - ♦ **JSON-BLOB.txt** is the name of your text file.
3. In the **JSON-BLOB.txt** file, delete the header information and array wrapper, leaving just the JSON blob. Confirm that the ID in this output matches **ZONE-ADMIN-CLIENT-ID**.

Your remaining JSON blob looks similar to the truncated sample below:

```
{
  "id": "demo",
  "subdomain": "demo",
  "config": {
    "clientSecretPolicy": {
      "minLength": -1,
```

```

    "maxLength": -1,
    "requireUpperCaseCharacter": -1,
    "requireLowerCaseCharacter": -1,
    "requireDigit": -1,
    "requireSpecialCharacter": -1
  },
  ...
},
"name": "demo",
"version": 2,
"description": "{\"plan_display_name\":\"demo\",
\"plan_description\":\"Demo Service Plan\"}",
"created": 1510116389000,
"last_modified": 1519859509000
}

```

4. In your `JSON-BLOB.txt` update the configurations in the JSON blob as needed, and then save the file.



WARNING: You must provide all `config` values, otherwise, data can be lost when doing an API update as a `PUT` command.

5. Submit a UAAC curl request to apply your updated configurations to the identity zone, as shown below.



WARNING: You must always run this command with the latest data by doing a `GET` before a `PUT` command.

```

uaac curl -k /identity-zones/ZONE-ADMIN-CLIENT-ID -X PUT
-H 'Content-Type: application/json' -d "$(cat file.txt)"

```

Where:

- `ZONE-ADMIN-CLIENT-ID` is your identity zone admin client ID created in [Create a UAA Identity Zone Admin Client](#) above.

A truncated example command would look similar to the following:

```

$ uaac curl -k identity-zones/demo\
-X PUT \
-H 'Content-Type: application/json' \
-d '{
  "subdomain": "demo",
  "config": {
    "clientSecretPolicy": {
      "minLength": 0,
      "maxLength": 255,
      "requireUpperCaseCharacter": 0,
      "requireLowerCaseCharacter": 0,
      "requireDigit": 0,
      "requireSpecialCharacter": 0
    },
    ...
  },
  ...
},

```

```

    "name": "demo",
    "version": 0,
    "description": "{\\"plan\\_display\\_name\\":\\"demo\\",
    \\"plan\\_description\\":\\"Demo Service Plan\\"}",
    "created" : 1529690485998,
    "last_modified" : 1529690485998
  }

```

For a full list of UAA API update parameters, see the [Identity Zones Update Documentation](#).

Modify Branding

You can optionally modify the branding of your login page by changing your company name, logos, legal text, and legal links.

Using the steps in [Update UAA Identity Zone Configurations with the API](#) above to retrieve the identity zone configurations for your SSO plan, add or modify the branding section according to the [UAA API documentation](#).

An example branding section is shown below.



Note: All values are optional. You can also generate the base64 text of your PNG images using commands, such as `base64 image.png`.

```

"branding": {
  "companyName": "Pivotal",
  "productLogo": "(base64 of png image here, will show up as image on login page)"
,
  "squareLogo": "(base64 of png image here, will show up as browser icon)",
  "footerLegalText": "©2017 Pivotal Software, Inc. All Rights Reserved.",
  "footerLinks": {
    "Privacy Policy": "https://run.pivotal.io/policies/privacy-policy/",
    "Terms of Service": "https://run.pivotal.io/policies/terms-of-service",
    "Up to three links, label here": "https://link-here"
  }
},

```

Add Default Groups for Users

Optionally, you can add additional default groups for all users. You do not need to do manual group assignment or group mappings for these groups. Use default groups only for universal scopes that all users can have, such as for a global read-only resource.

Using the steps in [Update UAA Identity Zone Configurations with the API](#) to retrieve and update the current identity zone configurations for your SSO plan, update the default groups section according to the [UAA API documentation](#).

An example of the default groups section is shown below. You can add more groups in the array list.

Users will automatically have these scopes though they are not explicitly assigned to users.

```
"userConfig": {
  "defaultGroups": [
    "openid",
    "password.write",
    "uaa.user",
    "approvals.me",
    "profile",
    "roles",
    "user_attributes",
    "uaa.offline_token",
    "new.group.everyone.should.have",
    "another.new.group.everyone.should.have"
  ]
},
```

Rotate JSON Web Token (JWT) Signing Keys

To rotate JWT signing keys:

1. Generate a private key that can be used for signing. For example, run `ssh-keygen -t rsa`.

Generate your signing keys in a secure manner. Refer to your security organization for acceptable key generation practices.

2. Take the value of the generated private key and make it a single line of text, replacing all new lines with `\n`. For example:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEA63iy3EpQG46eRzUKpI8sB/AQdbZwwrDkfPGg5Xt5xNM/wQrO
5l/yWp3lCElSqnKpJbCGu1DQThB47kGQjBoXL8TcrkxuCyuxaV7B5ryq3w+g3R1x -----END RSA
PRIVATE KEY-----
```

Becomes:

```
-----BEGIN RSA PRIVATE KEY-----\nMIIEogIBAAKCAQEA63iSAMPLEzUKpI8sB/AQdbZwwrDkSA
MPLEt5xNM/wQrO\n5l/yWp3lCElSqnKSAMPLE8TcrkxuCyuxaV7B5ryq3w+g3R1x\n-----END RSA
PRIVATE KEY-----\n
```

3. Using the steps in [Update UAA Identity Zone Configurations with the API](#) above to retrieve and update the identity zone configurations for your SSO plan, update the token policy section to add your new generated private key as the value for `signingKey`.

An example of this section is shown below.



Note: After you begin to configure JWT signing keys within a service plan, you can no longer default to share the multi-tenant JWT signing key inherited from the default zone.



Note: The first time you set a signing key for an identity zone, existing issued tokens are immediately invalidated for online validation. You may need to restart applications that do offline validation for the new signing keys to take effect.

```
"tokenPolicy": {
  "accessTokenValidity": -1,
  "refreshTokenValidity": -1,
  "jwtRevocable": false,
  "refreshTokenUnique": false,
  "refreshTokenFormat": "jwt",
  "activeKeyId": "first-signing-key",
  "keys" : {
    "first-signing-key" : {
      "signingKey" : "-----BEGIN RSA PRIVATE KEY-----\nMIIEogIBAAKCAQE63
iSAMPLEzUKpI8sB/AQdbZwwrDkSAMPLEt5xNM/wQrO\n5l/yWp3lCElSqnKSAMPLE8TcrkxuCyuxaV7
B5ryq3w+g3R1x\n-----END RSA PRIVATE KEY-----\n"
    }
  },
}
```

For more information, see [UAA API documentation](#).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Identity Providers



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 Application Service 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.

- Under **Password Policy Settings**, select **Use Recommended Settings**, **Use Default Settings**, or enter custom settings in the fields below.
- 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 (UAA) as follows:

- If you do not already have the UAA installed, run `gem install cf-uaac` in a terminal window.
- 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`
- Record the **App ID** and **App Secret**. These are used as your client ID and client secret.
- 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
```

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

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

- When prompted with `Client secret`, enter the **App Secret** admin client secret recorded above.
- 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 Application Service 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 need 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 Application Service 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

Follow the steps below to add an external SAML identity provider:

1. Log in to the SSO Operator Dashboard at <https://p-identity.YOUR-SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your Pivotal Application Service 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 Application Service 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**. You may also optionally configure the **Issuer**, **User Info Endpoint URL**, or **Response Type** if required for your OIDC Identity Provider integration.
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.



WARNING: Pivotal recommends against reusing LDAP service accounts across environments. LDAP service accounts should not be subject to manual lockouts, such as lockouts that result from users utilizing the same account. Also, LDAP service accounts should not be subject to automated deletions, since disruption to these service accounts could prevent user logins.

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 Application Service 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:
 1. Enter the **Hostname** and **Port**.
 2. Select the applicable **Security protocol**.
 3. Select the applicable **Referral**.
 4. Enter the **User DN** and **Bind Password** for your LDAP service account.
 5. Under the **Users** section, enter the **Search Base**.
 6. Under the **Users** section, you may also enter in **Search Filter** (Optional).
 7. 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.
 8. Under the **Groups** section, you may enter the **Search Base** (optional) and **Search Filter** (optional) to associate LDAP groups with your user. If you want 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 Application Service 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 Allowlist for an External Identity Provider

An administrator can include groups from an external identity provider in a group allowlist. The list of groups in the allowlist 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 allowlist containing external groups, the app must request the `roles` scope, and the group allowlist 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 Application Service 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**.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Enabling Identity Provider Discovery



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates,

 upgrade to a supported version.

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:
 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 Application Service tile in Ops Manager under the Credentials tab.
 2. Click the plan name and select **Configure** under the plan menu.
 3. Select the checkbox under the **Identity Provider Discovery** section and click **Save**.

The screenshot shows the 'Single Sign-On' configuration interface. The breadcrumb trail is 'Plans > Acme INC > Configure'. The page title is 'Configure'. Under the 'Token Policy' section, there are two input fields for 'Access Token Expiration' and 'Refresh Token Expiration', both with a 'Seconds' label. Below each field is a checked checkbox for 'Use System Default' (12 hours for Access Token, 30 days for Refresh Token). Under the 'Identity Provider Discovery' section, there is a checked checkbox for 'Enabled'. A 'Save' button is at the bottom.

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.

The screenshot shows the 'Single Sign-On' configuration interface for the 'Internal User Store' tab. The breadcrumb trail is 'Plans > Acme INC > Identity Providers > Internal User Store'. The page title is 'Internal User Store'. Under the 'Email Domains' section, there is a text input field with the value 'company.com, supplier-1.com, supplier-2.com'. Below this is a section for 'Authentication Policy' with two unchecked checkboxes: 'Disable Internal Authentication' and 'Disable User Management'. Below that is a 'Password Policy Settings' section with two buttons: 'Use Recommended Settings' and 'Use Default Settings'. At the bottom, there are four input fields for 'Password Complexity' and 'Lockout Policy': 'Min Length' (1), 'Uppercase' (0), 'Failures Allowed' (5), and 'Lockout Period' (300).

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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Managing Users



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 Application Service 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.

Internal Users [Invite User](#)

Search for Users

Enter a username [Search](#)

[Resend Invite](#) [Reset Password](#) [Delete User](#)

<input type="checkbox"/>	Username	Name	Verified	Last Modified
<input type="checkbox"/>	test@test.com		No	5/15/2017 12:02:49 PM

- **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.

test@test.com [Resend Invite](#)

[Profile](#) [Permissions](#)

Email **NOT VERIFIED**
test@test.com

First Name

Last Name

Phone Number

[Delete](#) [Cancel](#) [Save User](#)

- **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.

test@test.com [Resend Invite](#)

[Profile](#) [Permissions](#)

User does not have any permissions [Select Permissions](#)

[Delete](#) [Cancel](#) [Save User](#)

- **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 Application Service 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.

Okta SSO Users

Search for Users

Enter a username

Search

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.

Okta SSO Users

Search for Users

Enter a username

Search

Delete User

	Username	Name	Verified	Last Modified
<input type="checkbox"/>	tiwang+test@pivotal.io	Test User	Yes	5/17/2017 7:51:06 PM

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

tiwang+test@pivotal.io

Profile

Permissions

Username

tiwang+test@pivotal.io

Email

tiwang+test@pivotal.io

VERIFIED

First Name

Test

Last Name

User

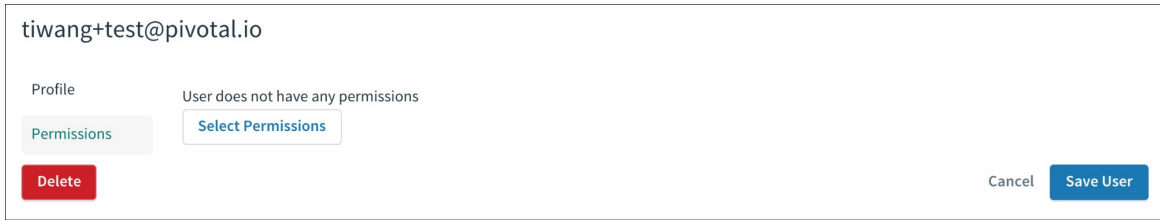
Phone Number

Delete

Cancel

Save User

- **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

You may also use the UAA CLI, or [UAAC](#), to manage users for the SSO service. You can use this approach to programmatically 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 context` command to display the client context. Verify that you have the sufficient `scim.write` or `scim.create` permissions under the `scope` section.

```
$ uaac context

[1] * [admin]
```

```

client_id: MyAdminAppId
access_token: aBcdEfg0hIJKlm123.e
token_type: bearer
expires_in: 43200
scope: scim.read scim.write
jti: 91b3-abcd1233

```

- 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
```

- 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
```

- 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
```

[Create a pull request or raise an issue on the source for this page in GitHub](#)

About Performance



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic lists resources where you can learn about evaluating the performance of the Single Sign-On (SSO) service for Pivotal Cloud Foundry (PCF).

As an operator, you need to know about the performance of logins and token flows. For example, you can monitor how long it takes for UAA to issue the tokens that allow users to log in to apps.

The SSO service relies on the performance of User Account and Authentication (UAA) component of Pivotal Application Service.

The following table contains links to topics that describe how to monitor and interpret UAA performance.

For more information about...	See...
UAA in general	Component: User Account and Authentication (UAA) Server
UAA performance	UAA Performance
UAA performance metrics	UAA Performance Metrics
UAA metrics	UAA Metrics

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Backing Up and Restoring



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how to back up and restore the Single Sign-On (SSO) service for Pivotal Cloud Foundry (PCF).

To back up and restore the SSO service, use the BOSH Backup and Restore (BBR) tool to back up and restore PCF. For more information, see [Backing Up and Restoring Pivotal Cloud Foundry](#).

As part of backing up and restoring PCF, BBR also backs up and restores the SSO service tile information. The information is included in the Cloud Controller (CC) and User Account and Authentication (UAA) data of Pivotal Application Service (PAS).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Developer Guide

Determining SSO Application Type



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic explains how to determine your Single Sign-On for PCF app 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, its type is Service-to-Service App.

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

App Type	Single Sign-On App Type	OAuth Grant Type
Web	Web App	Authorization Code
Native Mobile, Desktop, or Command Line	Native Mobile App	Resource Owner 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. The Native Mobile App application type works only with back-channel protocols, such as internal UAA store or LDAP. It does not work with front-channel protocols, such as SAML.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

OAuth 2.0 Grant Types



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic explains how OAuth 2.0 grant types work with different app types.

Authorization Code Grant Type

The authorization code grant type is the most commonly used grant type. This grant type is for server-side apps.

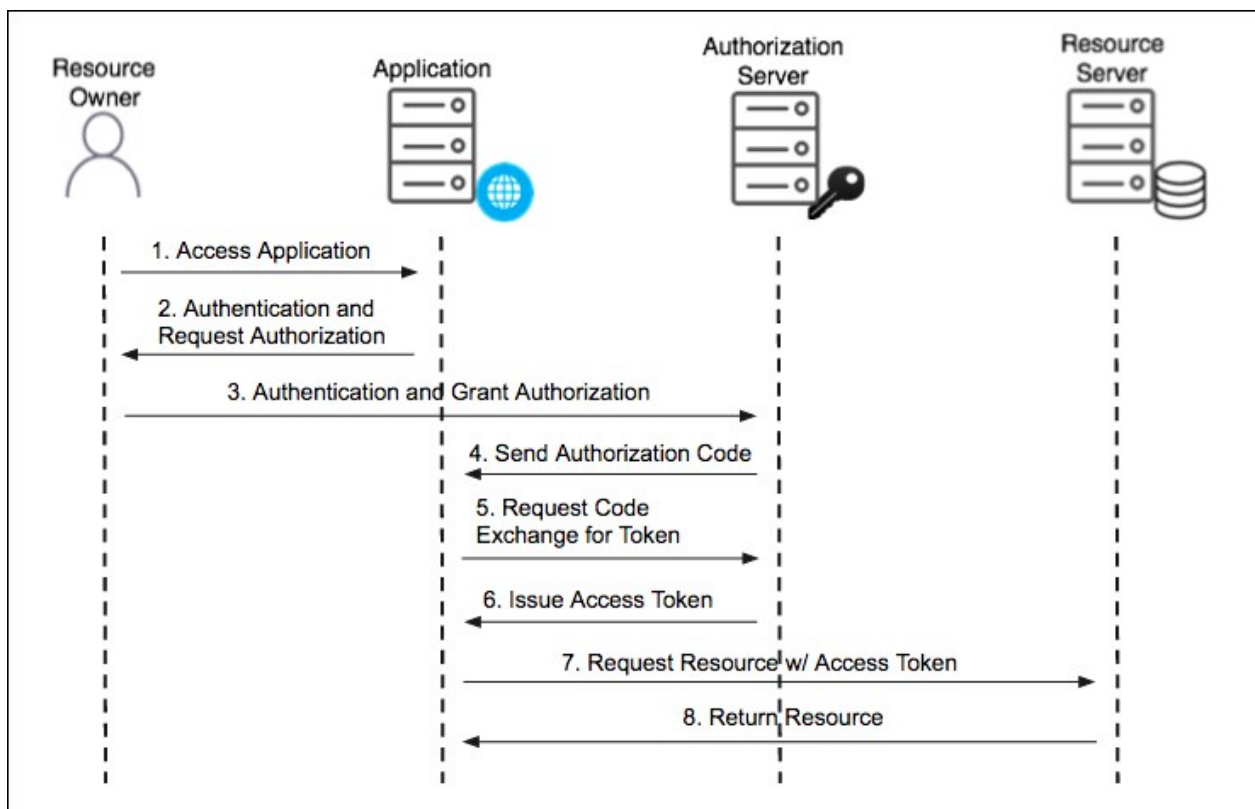
Authorization Code Grant Type Roles

The Authorization Code grant type uses the following 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 apps 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. Apps access the server through APIs.

Authorization Code Grant Type Flow

The following diagram shows the flow for the Authorization Code grant type:



1. **Access Application:** The user accesses the app and triggers authentication and authorization.
2. **Authentication and Request Authorization:** The app redirects the user to the authorization server where it prompts the user for their username and password. The first time the user

goes through this flow for the app, the user sees an approval page. On this page, the user can choose permissions to authorize the app to access resources on their behalf.

3. **Authentication and Grant Authorization:** The authorization server receives the authentication and authorization grant.
4. **Send Authorization Code:** After the user authorizes the app, the authorization server sends an authorization code to the app using a redirect.
5. **Request Code Exchange for Token:** The app uses the authorization code to request an access token from the authorization server. This gives the app 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 app 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 app to receive.

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

Client Credentials Grant Type

This grant type is for apps that can request an access token and access resources on its own. These apps often use services that call APIs without users.

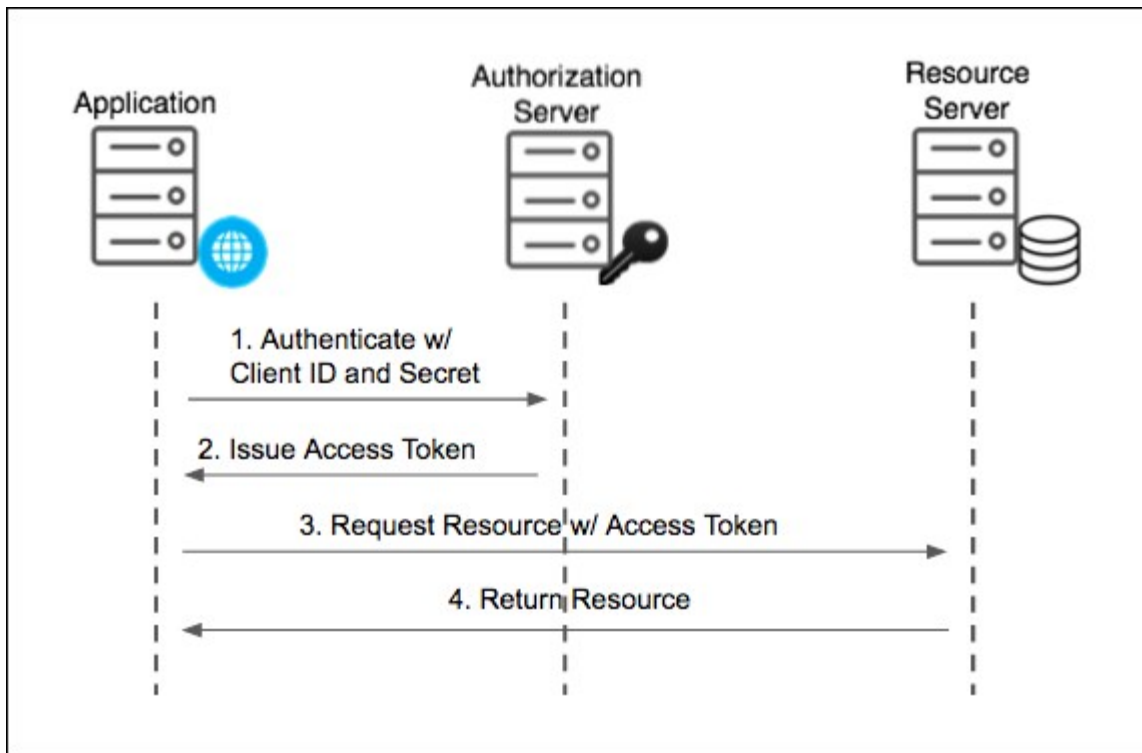
Client Credentials Grant Type Roles

The Client Credentials grant type uses the following roles:

- **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 apps 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. Apps access the server through APIs.

Client Credentials Flow

The following diagram shows the flow for the Client Credentials grant type:



1. **Authenticate w/ Client ID and Secret:** The app 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 app 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 app.

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

Resource Owner Password Grant Type

For Native Mobile and Desktop apps, Single Sign-On for PCF supports the Resource Owner Password OAuth 2.0 grant type. This password grant type is for highly trusted apps where resource owners share their credentials directly with the app.

Resource Owner Password Grant Type Roles

The Resource Owner Password grant type uses the following roles:

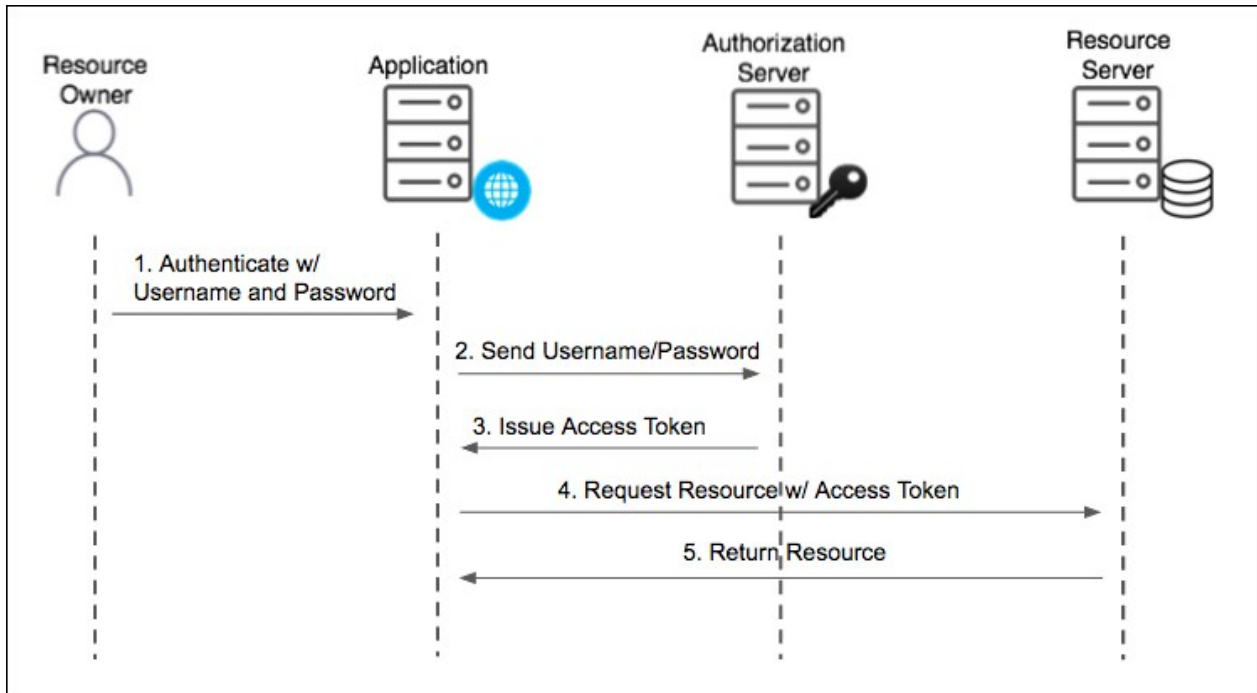
- **Resource Owner:** A person or system capable of granting access to a protected resource.
- **Application:** A client that makes protected requests using the authorization of the resource owner.
- **Authorization Server:** The Single Sign-On server that issues access tokens to client apps

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. Apps access the server through APIs.

Resource Owner Password Flow

The following diagram shows the flow for the Resource Owner Password grant type:



1. **Authenticate w/ Username and Password:** The user authenticates with the app using their username and password.
2. **Send Username/Password:** The app 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 app 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 app to receive.

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

Implicit Grant Type

The Implicit grant type is for apps with a client secret that is not guaranteed to be confidential.

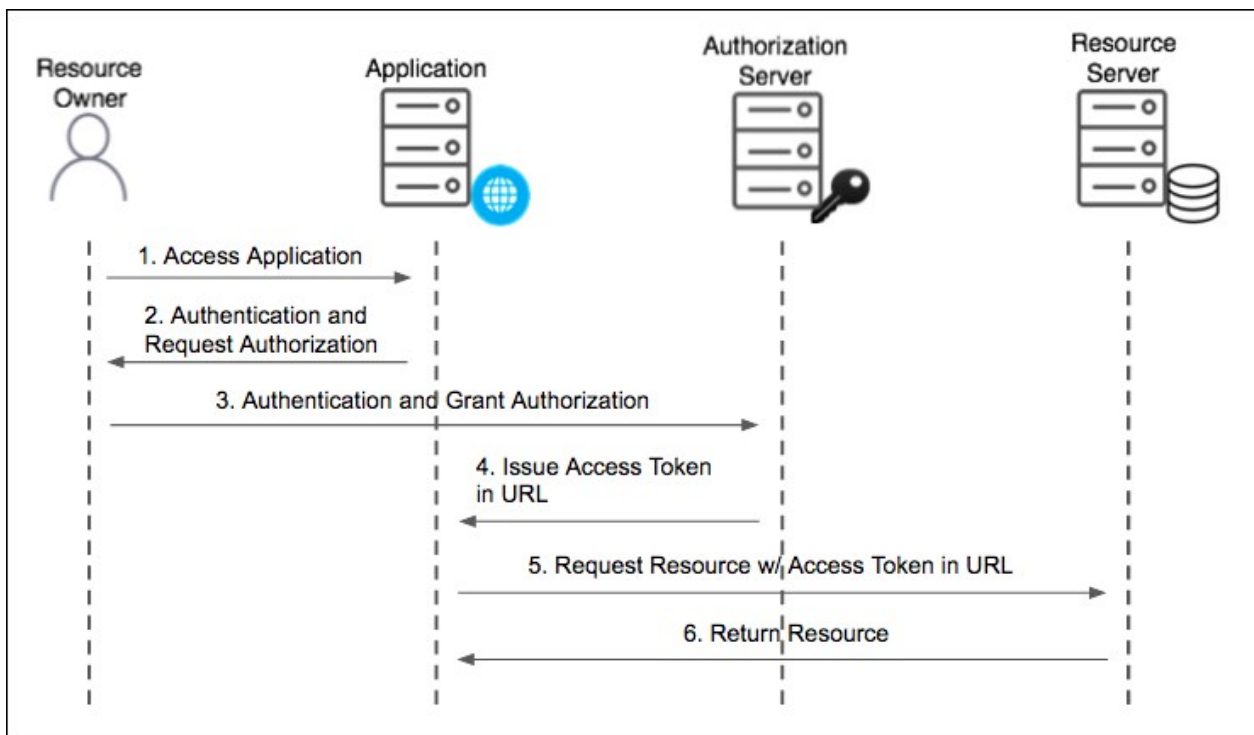
Implicit Grant Type Roles

The Implicit grant type uses the following 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 apps 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. apps access the server through APIs.

Implicit Flow

The following diagram shows the flow for the Implicit grant type:



1. **Access Application:** The user accesses the app and triggers authentication and authorization.
2. **Authentication and Request Authorization:** The app prompts the user for their username and password. The first time the user goes through this flow for the app, the user sees an approval page. On this page, the user can choose permissions to authorize the app 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 app 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 app to receive.

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Common App Architecture Patterns



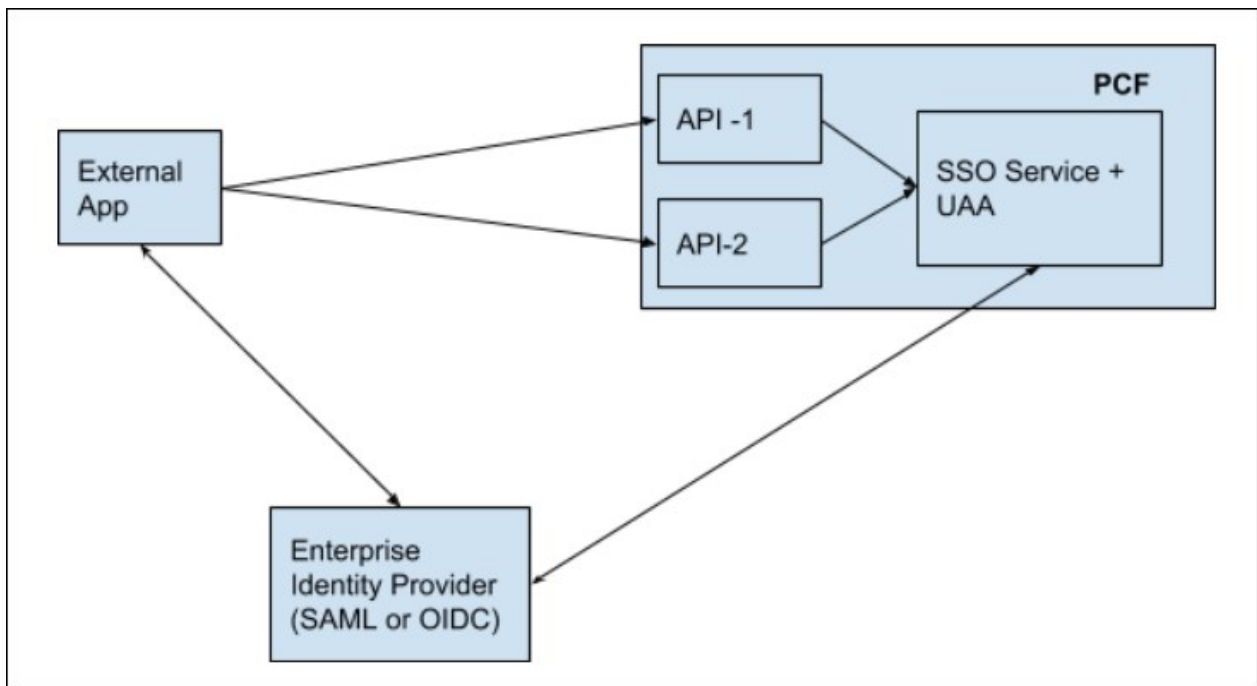
Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes common app architecture patterns for enterprise developers.

External Apps Calling into PCF APIs

This section describes common architecture patterns for authenticating external apps calling into Pivotal Cloud Foundry (PCF) APIs. In these patterns, external apps secured either by Security Assertion Markup Language (SAML), or by OpenID Connect (OIDC) providers using JSON Web Tokens (JWTs), interact with Pivotal SSO to gain access to PCF services.

The following diagram is a conceptual view of an external app protected by a SAML or OIDC enterprise identity provider (IDP).



In this diagram:

- You have an external app running outside PCF.
- The external app is secured by an enterprise SAML or OIDC identity provider, that is, users

authenticate via the SAML or OIDC provider into the external app.

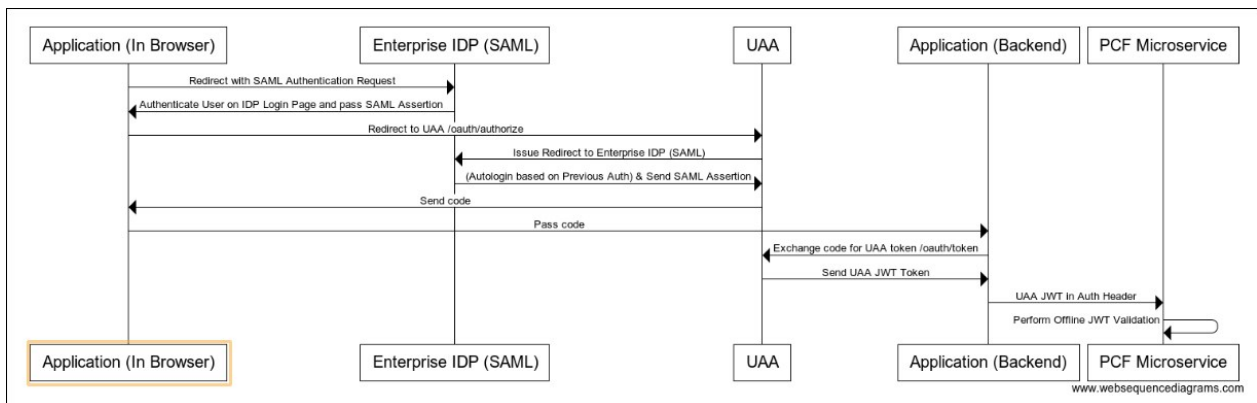
- The external app needs to invoke API-1 and API-2, which are Spring Boot microservices running on PCF.
- API-1 and API-2 are protected by the SSO service using OAuth.

The following sections describe three authentication models that can be used for external apps using SAML or OIDC enterprise IDPs to call into PCF APIs:

- [UAA Authorization Code Grant](#)
- [SAML Bearer Token Exchange](#)
- [JWT Exchange](#)

UAA Authorization Code Grant — Browser

The following sequence diagram illustrates the UAA authorization code grant model. This diagram shows a SAML flow, but the interactions between the app, enterprise IDP, and UAA can also use an OIDC enterprise IDP. In that case, JWT ID tokens replace SAML and SAML assertions.

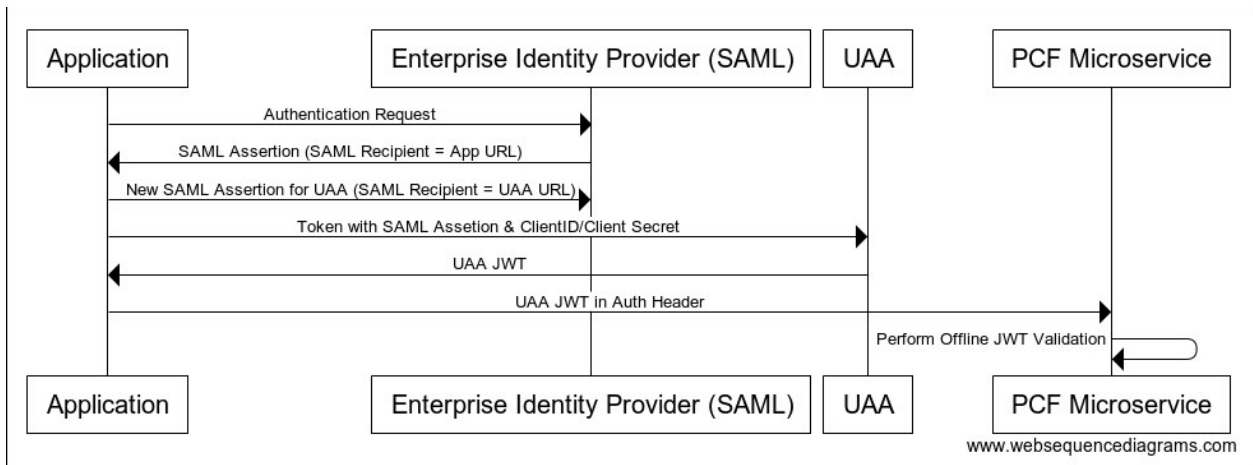


In this model:

- The authenticated user's session against the enterprise IDP is leveraged, which makes direct user authentication against the UAA transparent to the user.
- The user/browser calling a PCF microservice is redirected to UAA and then to the IDP. Because this user session with the IDP is already authenticated, the redirect back to UAA and then back to the PCF microservice is transparent to the user.
- Your enterprise IDP applies any security policies defined by your corporate policy, for example, multi-factor authentication (MFA), Kerberos, or PKI certificates.
- Behind the scenes the UAA authenticates the user via SAML, because the user has a logged in session with the IDP.
- UAA then generates a JWT for the authenticated user.

SAML Bearer Token Exchange — Back End

The following sequence diagram illustrates the SAML bearer token exchange model.



In this model:

- The user has already authenticated against the enterprise IDP via SAML through interactions with the existing integration with the IDP upstream.
- The authenticated session against the IDP empowers the user to get a SAML assertion intended for the Pivotal SSO (also known as UAA) audience. In the SAML assertion, the recipient and audience must match UAA. See the [Example SAML Assertion](#) below.
- The UAA allows for a token exchange permitting the upstream caller to get a UAA JSON web token (JWT), which authorizes access to PCF hosted microservices, leveraging the SAML assertion mentioned above.

Example SAML Assertion

In the SAML assertion used in the SAML bearer token exchange, the **Recipient** and **Audience** must match UAA. For example:

Recipient

The **Recipient** must match the UAA entity ID.

```
Recipient="https://demo.login.uaa-example.com/oauth/token/alias/demo.login.uaa-example.com"/>
```

Audience

The **Audience** must match the UAA assertion consumer service URL.

```
<saml2:AudienceRestriction><saml2:Audience>demo.login.uaa-example.com</saml2:Audience>
</saml2:AudienceRestriction>
```

Example SAML Assertion

The **Recipient** and **Audience** properties are located near the bottom of the example SAML assertion below.

```
<?xml version="1.0" encoding="UTF-8"?><saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" ID="id41261893195735352003413868" IssueInstant="2018-01-24T19:23:15.522Z" Version="2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"><saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">http://www.okta.com/exk9sgb2ayyAikL150h7</saml2:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:SignedInfo><ds:CanonicalizationMet
```

```

hod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><ds:SignatureMethod Algorithm
="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/><ds:Reference URI="#id4126189319
5735352003413868"><ds:Transforms><ds:Transform Algorithm="http://www.w3.org/2000/09/xm
ldsig#enveloped-signature"/><ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc
-c14n#"/><ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" Pre
fixList="xs"/></ds:Transform></ds:Transforms><ds:DigestMethod Algorithm="http://www.w3
.org/2001/04/xmlenc#sha256"/><ds:DigestValue>oTsVYrWJ4Yah1eM3p0e4DCLP3NlsgFoAZ6R/KIIon
L8=</ds:DigestValue></ds:Reference></ds:SignedInfo><ds:SignatureValue>BYg9pYj2MwO4OvQv
tuH2WOWemcEew7R6dIyxEaUC9sAtTyMBB0dumMhDZMtOXu7G6+Uoba7B1XqAS8YM5/1lQsW/oGZAH9NuhYhNW1
eWw8eSrTQjpfKn61Vei2EmihWwTRptBZUucu4ZSblvqPnUYt0SF/hMfHqYRbILeicZTgT/Tl1lOIMoPcET7JHC
1ZkMYGJfKjXue1t34FER55ce1CnQQIXBN435R0WWLhx0UND9XGWP1B3ddtaMJleh09EZDECE1ORGP1niVp1LSs
x0QE1inVTr7Qn7+x3tG1X/9MVgEDevZaGdZzwdbkwwfDssFWppjLpqpCBLZLK8USSN1Q==</ds:SignatureVa
lue><ds:KeyInfo><ds:X509Data><ds:X509Certificate>MIIDpDCCAoygAwIBAgIGAVmywjWWMA0GCSqGS
Ib3DQEBBQUAMIGSMQswCQYDVQQGEwJVUzETMBEG
A1UECAwKQ2FsaWZvcmlkZXRhZARBgNVBAMCMRldi0yODEzNzYxHDAaBgkqhkiG9w0BCQEW
MBIGA1UECwwLU1NPUHJvdmlkZXRhZARBgNVBAMCMRldi0yODEzNzYxHDAaBgkqhkiG9w0BCQEW
DWluZm9Ab2t0YS5jb20WbHcNMTCwMTE4MTgwNTI5WhcNMjcwMTE4MTgwNjI5WjCBkjELMAKGA1UE
BhMCVVMwEzARBgNVBAGMCkNhbm3JuaWEwEzFjAUBGNVBACMDVNhbiBGcmFuY2l2Y28xDTALBgNV
BAoMBE9rdGEzFDASBgNVBASMC1NTT1Byb3ZpZGVyMRRwEQYDVQQDDApkZXYtMjgxMzc2MRwwGyYJ
KoZIHvcNAQkBFglpbmZvQG9rdGEuY29tMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
0okZSvNaxO/QzpT92pxALWewO1j3F0DyFRjWz1x4u8AbPjJDizbr42pnm/dOxw5bij2CecvIvI3b
G/LNMh0NMB1uuMwRppIpNkU0mu/8b3ulszmGMSULRIAtCQFIKAd8VXApbmNlLsfzN5CnJzeDEZ29
3E/RGVr0WvSUKWYZaij57BfH2r2A44TZfRNPfUgtsvsVVQvtwDgKBo+rNqZIkQoMi0hdpX2Z522Z
16vpbDGu56kWR0fqyfoshKHPnHNvk/c0HkwcKCIAM117DW95PnrTxjx7QQuLZibUFPD1sQE2S4Vxe
W6kxXhT8ttmML0OjirEqtD+98BcbqCc9SgYtzwIDAQABMA0GCSqGSIB3DQEBBQUAA4IBAQDKqs49
VBGPRTAWvGm+giBHT2uJd5JCefE6ap/OPp+ajfslXXH3yU0q6CiyKliVgS9j15MOVBDTou8vTtsK
w0TmdG1NHKJCjqpTe2h/+3uvCG2yv9D6rfDiQcO4KgeG+5hXnS2fGcFTuCuMODX7ivEYB9eeAqXkJG
4LFwxVhse8j0rwdkPESkdL7KdThzK5rsM3tWihSsuccm4a6Zp6faFZzWhvd6ujBGilLtaVHP9jUG
eMHVqMYK6C91CalL4/kGUJYGsKhbuF4CdjlWk9PB4PvNLn+ijWk9dYkVlQYMH93Lg9T/2OYVBux7
MQsY0xtKYytwky+LiElSjODZPQvYXaS3</ds:X509Certificate></ds:X509Data></ds:KeyInfo></ds:S
ignature><saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"><saml2:Name
eID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">person1@company.co
m</saml2:NameID><saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bear
er"><saml2:SubjectConfirmationData NotOnOrAfter="2018-01-24T19:28:15.522Z"
Recipient="https://demo.login.uaa-example.com/oauth/token/alias/demo.login.uaa-example
.com"/>
</saml2:SubjectConfirmation></saml2:Subject><saml2:Conditions NotBefore="2018-01-24T19
:18:15.522Z" NotOnOrAfter="2018-01-24T19:28:15.522Z" xmlns:saml2="urn:oasis:names:tc:S
AML:2.0:assertion">
<saml2:AudienceRestriction><saml2:Audience>demo.login.uaa-example.com</saml2:Audience>
</saml2:AudienceRestriction>
</saml2:Conditions><saml2:AuthnStatement AuthnInstant="2018-01-24T19:23:15.522Z" Sessi
onIndex="id1516821795522.1919419636" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertio
n"><saml2:AuthnContext><saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:clas
ses:PasswordProtectedTransport</saml2:AuthnContextClassRef></saml2:AuthnContext></saml
2:AuthnStatement><saml2:AttributeStatement xmlns:saml2="urn:oasis:names:tc:SAML:2.0:as
sertion"><saml2:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname
-format:basic"><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns
:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">person1@company.
com</saml2:AttributeValue></saml2:Attribute><saml2:Attribute Name="fn" NameFormat="urn
:oasis:names:tc:SAML:2.0:attrname-format:unspecified"><saml2:AttributeValue xmlns:xs="
http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
" xsi:type="xs:string">Sree</saml2:AttributeValue></saml2:Attribute><saml2:Attribute N
ame="ln" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"><saml2:A
tttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:type="xs:string">Tummididi</saml2:AttributeValue></saml2:A
tttribute><saml2:Attribute Name="roles" NameFormat="urn:oasis:names:tc:SAML:2.0:attrnam
e-format:unspecified"><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema
" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Everyone<

```



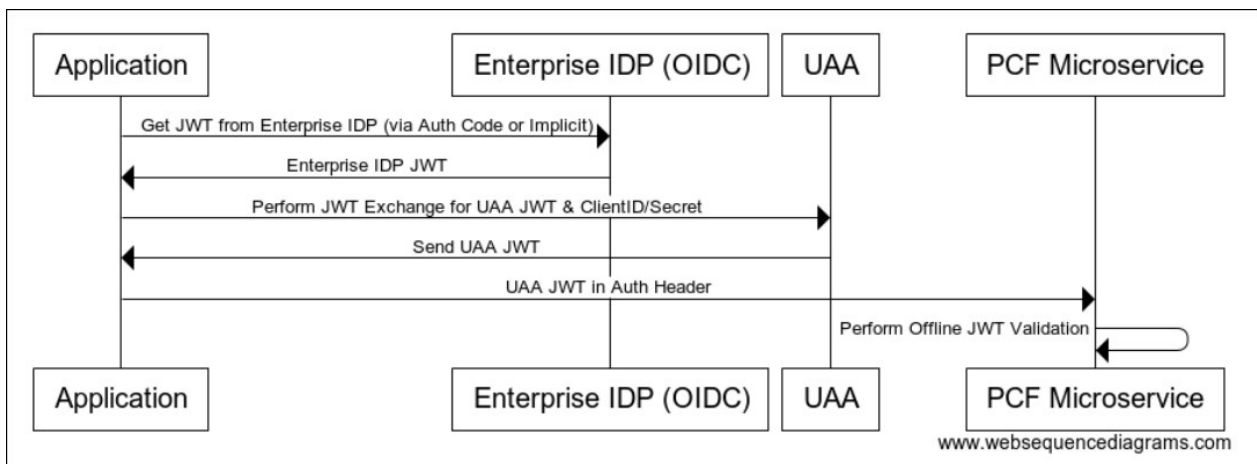
```

/saml2:AttributeValue><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Writer-Ad
min</saml2:AttributeValue><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSc
hema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Reade
r</saml2:AttributeValue><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSche
ma" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Writer<
/saml2:AttributeValue><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema
" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Writer-a
Admin</saml2:AttributeValue><saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XML
Schema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Rea
der-Admin</saml2:AttributeValue></saml2:Attribute></saml2:AttributeStatement></saml2:A
ssertion>

```

JWT Bearer Token Exchange

The following sequence diagram illustrates the JWT bearer token exchange model. This flow is for external apps using OIDC.



This model is similar to the SAML bearer token exchange flow:

- The upstream app contacts UAA and requests a PCF-native JWT. The app provides a JWT generated by the enterprise IDP as evidence that the user has been authenticated.
- The returned PCF-native JWT can then be used to invoke protected microservices hosted within PCF.

The difference between this flow and the SAML exchange one is that there is no need to get a specific SAML assertion for the UAA audience.

Example JWT Token Content

```

{
  "sub": "mysub",
  "iss": "https://my.idp.com",
  "aud": "http://appliesto/myidpjwt",
  "iat": 1517486551,
  "exp": 1517490151,
  "sess": "bf8d6812-0747-11e8-a94b-005056be1e86",
  "groups": [
    "my-admins"
  ],
  "emailAddress": [

```

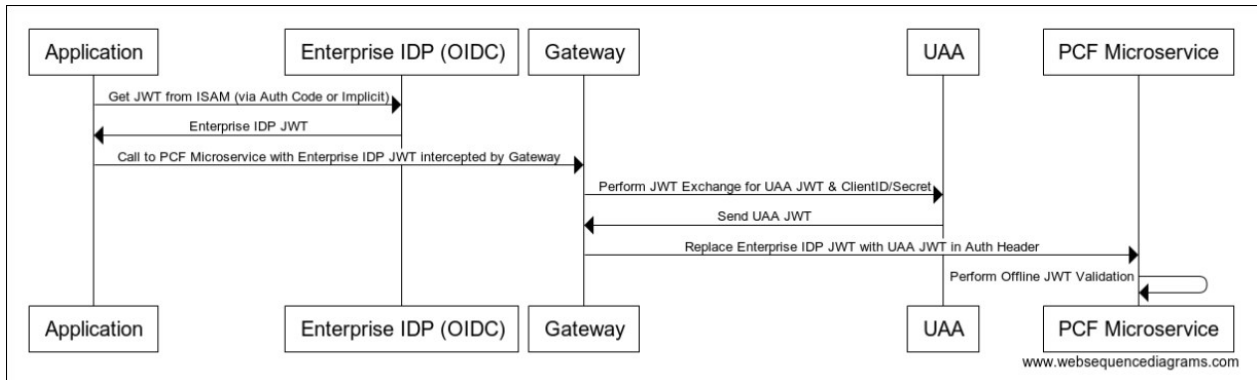
```

    "person1@company.com"
  ]
}

```

Handling the JWT Token Exchange Using a Gateway

In the sequence illustrated above, the token exchange is handled by the app. However, you can abstract away the token exchange from the app. Instead, a gateway such as Apigee or Mulesoft intercepts the call and handles the token exchange transparently, as shown in the diagram below.



Set Up for SAML or JWT Bearer Token Exchange

The following setup procedure uses the example shown in the [conceptual diagram](#) above.

1. Create API-1 and API-2 as resources in the SSO service, for example, `api1.read`, `api1.write` and `api2.read`, `api2.write`.

For instructions, see [Create or Edit Resources](#).

2. Create an Admin Client with the ability to create more OAuth Clients under the SSO Service plan. Therefore, use the scope `clients.admin`.

For instructions, see [Create Admin Client](#).

3. Use UAAC to target the SSO Service Plan and log in using the Admin Client created in the previous step.

For instructions, see Step 4 in [Manage Users with the UAA CLI \(UAAC\)](#).

4. For each client of the external app, run the following command to register the client:

```
uaac client add -i
```

In this example, there are two clients: API-1 and API-2.

5. When prompted:
 1. Specify a client ID and secret, and record them for future use in the API call.
 2. Specify the Grant Type as either of the following:

- For SAML: `urn:ietf:params:oauth:grant-type:saml2-bearer`
 - For JWT: `urn:ietf:params:oauth:grant-type:jwt-bearer`
3. Specify the scopes. In this example, they are either: `api1.read` and `api1.write`, or `api2.read` and `api2.write`.
 4. You can leave the redirect URI and other options empty.
 6. A plan admin must do the following in the SSO Plan Administrator Dashboard:
 1. Add the enterprise SAML or OIDC (for JWT) provider.

For instructions, see [Add an External Identity Provider](#).

2. Do one of the following:
 - Set up external group mappings for `api1.read` and other scopes.

For instructions, see [Create or Edit Resource Permissions](#).
 - Add the corresponding scopes to the users that require this access.

For instructions, see [Managing Users](#).
 - Make the scopes as default authorities so that users do not need to be assigned the groups individually.

For instructions, see [Add Default Groups for Users](#).
7. The external app, or gateway, must make the following API call:
 1. Invoke the `/oauth/token` endpoint of the SSO plan with the parameters laid out in this documentation:
 - For SAML: [SAML2 Bearer Grant](#)
 - For JWT: [JWT Bearer Token Grant](#)
 2. Pass in the following:
 - Client ID and client secret registered for the external app
 - The SAML assertion or JWT token from the external app

The response is a token that you must add in the authorization header when making a call to API-1 or API-2.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Managing Service Instances



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has

reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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**.

The screenshot shows the Pivotal Apps Manager interface. On the left is a dark sidebar with a navigation menu. The main content area is titled 'Single Sign-On' and includes a description of the service. Below this is a 'Configure Instance' form. The form contains three dropdown menus: 'Instance Name' (with the value 'MY-SERVICE-INSTANCE'), 'Add to Space' (with the value 'MY-SPACE'), and 'Bind to App' (with the value '[do not bind]'). At the bottom of the form are three buttons: 'Show Advanced Options', 'Cancel', and 'Add'.

6. From the **Add to Space** drop-down menu, choose a space for the instance. This space hosts your application. The default is `development`.
7. 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.
8. 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**.

- 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Apps



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

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. 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.

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 repository for examples of [bootstrapping](#) these values.



Note: These configurations are only applied when initially binding to the service instance. A subsequent `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.

Property Name	Description	Default
Name	Name of the app	(N/A - Required Value)
<code>GRANT_TYPE</code>	Allowed grant type for the app through the SSO service. Only one grant type per app is supported by SSO.	
<code>SSO_IDENTITY_PROVIDERS</code>	<p>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:</p> <ul style="list-style-type: none"> Uppercase letters are converted to lowercase letters. Spaces, periods, and underscores are converted to hyphens. Multiple hyphens are combined into a single hyphen. <p>For example, if your identity provider name is <code>example.com Provider</code>, the corresponding origin key is <code>example-com-provider</code>.</p>	<code>uaa</code>
<code>SSO_REDIRECT_URIS</code>	Comma-separated allowlist of redirection URIs allowed for the app. Each value must start with <code>http://</code> or <code>https://</code> .	(Always includes the app route)
<code>SSO_SCOPES</code>	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.	<code>openid</code>
<code>SSO_AUTO_APPROVED_SCOPES</code>	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)
<code>SSO_AUTHORITIES</code>	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 <code>scim.read</code> , <code>idps.write</code> cannot be bootstrapped and must be assigned by zone/plan administrators. This value is ignored for any grant type other than client credentials.	<code>uaa.resource</code>

<code>SSO_REQUIRED_USER_GROUPS</code>	Comma-separated list of groups a user must have in order to authenticate successfully for the app.	(No value)
<code>SSO_ACCESS_TOKEN_LIFETIME</code>	Lifetime in seconds for the access token issued to the app by the SSO service.	43200
<code>SSO_REFRESH_TOKEN_LIFETIME</code>	Lifetime in seconds for the refresh token issued to the app by the SSO service.	259200 (not used for client credentials)
<code>SSO_RESOURCES</code>	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 <code>todo.read</code> and <code>todo.write</code> , must be specified in the same application manifest. Currently you cannot specify additional permissions in the same top level permission, such as <code>todo.admin</code> , in additional application manifests.	(No value)
<code>SSO_ICON</code>	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)
<code>SSO_LAUNCH_URL</code>	App launch URL that will be used for the app on the Pivotal Account dashboard if show on home page is enabled.	(Application route)
<code>SSO_SHOW_ON_HOME_PAGE</code>	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 in the [identity sample apps](#) repository.

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.

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 at the initial service binding time. Subsequent cf push of the application will **NOT** update the configurations. You will either need to manually update the configurations via the SSO dashboard or unbind and rebind the service 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.

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.
 1. Select an existing SSO service instance from the drop-down menu and click **Bind**.
 2. Create a new service instance:
 1. Click **or add from Marketplace**.
 2. Select the **Single Sign-On** service under Services Marketplace.
 3. Select a Service Plan, then click **Select this plan**.
 4. Enter an **Instance Name**, select a space, select an app, then click **Add**.

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 to launch the SSO dashboard.
4. Click your app.
5. Specify a value in the **App Launch URL** field that you want to set as the address of your app.
6. Upload an app icon for your app.
7. 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.

8. 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.

9. If your Application Type is **Web App** or **Single-Page JavaScript App**, enter an allowlist 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.
10. 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 **uua.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
uua.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.

11. 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.
12. Click **Save Config**. The **Next Steps** page appears, describing the endpoints required for app integration. For more information, see [Integrate SSO with Apps](#) below.

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 an allowlist 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 *uua.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
<i>openid</i>	Provides access to make OpenID Connect requests
<i>user_attributes</i>	Provides access to custom attributes from an external identity provider
<i>roles</i>	Provides access to external groups from an identity provider
<i>uua.resource</i>	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 managing 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.

To create an admin client:

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.

- Click **New App**.
- Enter an **App Name**.
- Under **Select an Application Type**, select **Service-to-Service App**.
- Click **Select Scopes** and choose what actions the admin client can perform from the following **Admin Permissions**:

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

- Click **Create App**. You are given the option to view and download the **App ID** and **App Secret**. Download or make note of this information for use with other SSO procedures.

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:

- Log in to Apps Manager as a Space Developer.
- Select the space where your app is located.
- Under **Applications**, click the name of your app.
- On the Application page, click **Delete App**.
- 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:

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Managing Resources



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 repository](#).

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.
 1. Enter one or more **Attributes** or **Actions** for each permission.
 2. 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 Application Service 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 Application Service 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Integration Guides

Active Directory Federation Services Integration Guide

Active Directory Federation Services Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

Active Directory Federation Services

- Active Directory Federation Services subscription.
- A user with admin privileges.



Note: To configure SAML, you must have the Pivotal 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Active Directory Federation Services as an Identity Provider

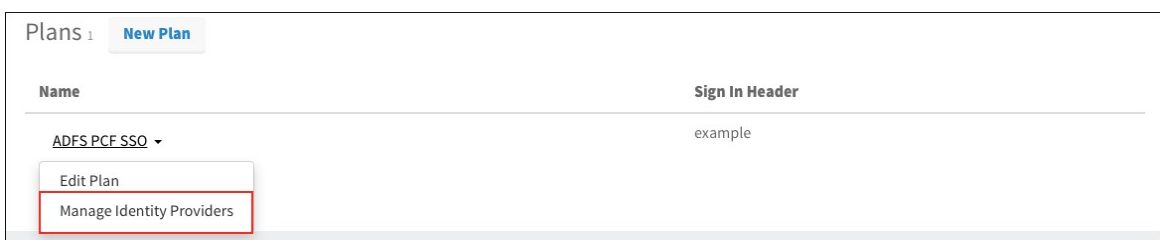


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

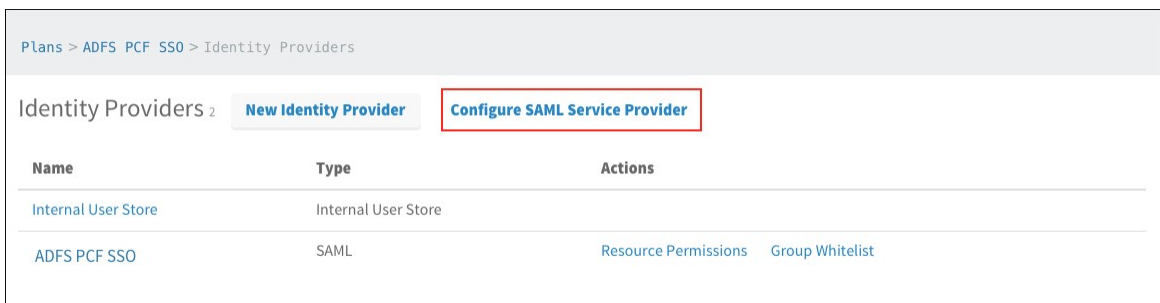
This topic describes how to set up Active Directory Federation Services (ADFS) as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and ADFS.

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.

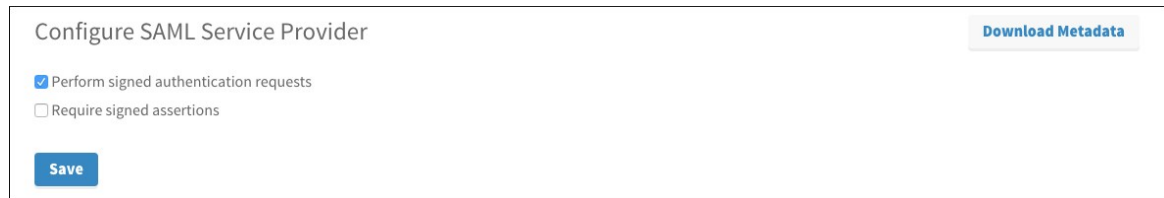


3. Click **Configure SAML Service Provider**.



4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key

signature and identity provider validation.



Configure SAML Service Provider

☒ Perform signed authentication requests

☐ Require signed assertions

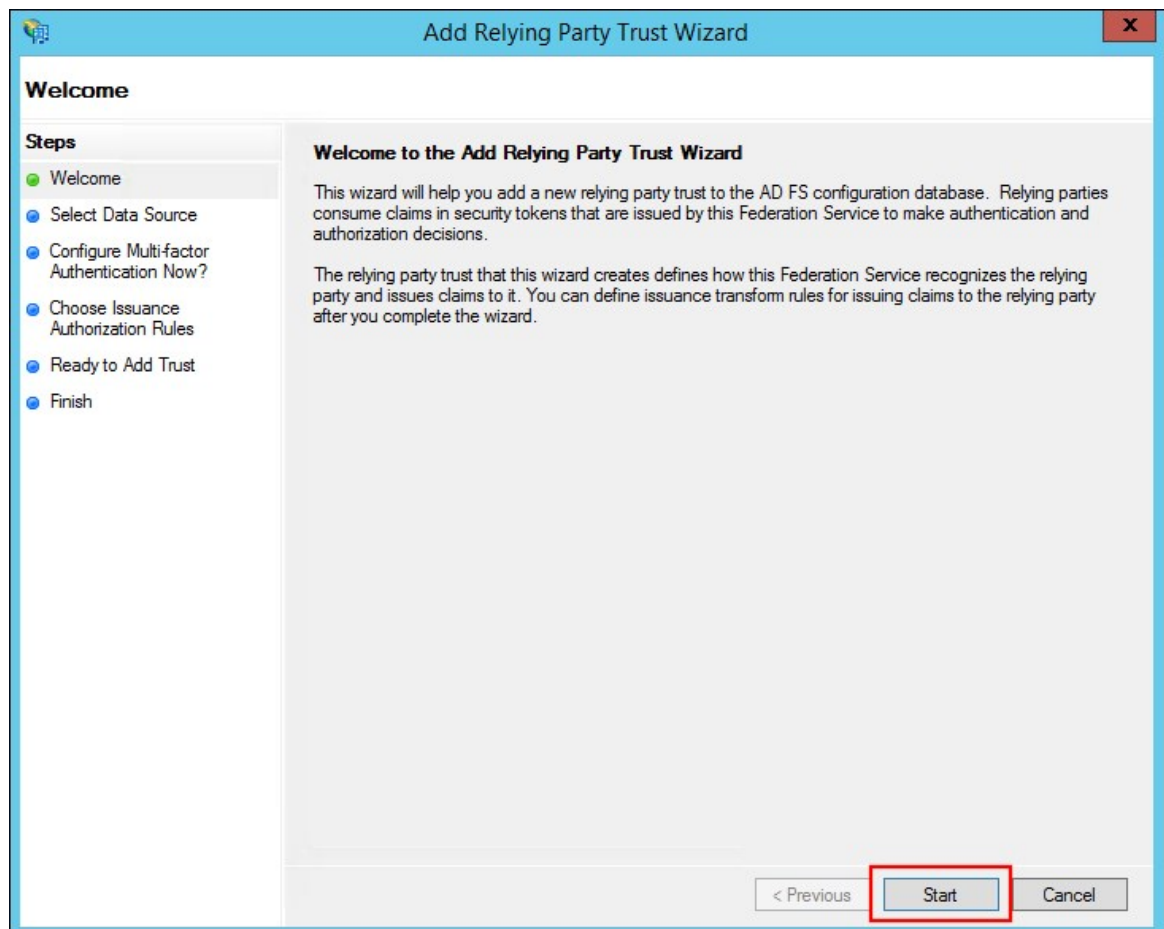
Save

Download Metadata

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**.



Add Relying Party Trust Wizard

Welcome

Steps

- Welcome
- Select Data Source
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Welcome to the Add Relying Party Trust Wizard

This wizard will help you add a new relying party trust to the AD FS configuration database. Relying parties consume claims in security tokens that are issued by this Federation Service to make authentication and authorization decisions.

The relying party trust that this wizard creates defines how this Federation Service recognizes the relying party and issues claims to it. You can define issuance transform rules for issuing claims to the relying party after you complete the wizard.

< Previous Start Cancel

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

Add Relying Party Trust Wizard

Select Data Source

Steps

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Select an option that this wizard will use to obtain data about this relying party:

☐ Import data about the relying party published online or on a local network

Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

Example: fs.contoso.com or https://www.contoso.com/app

☒ Import data about the relying party from a file

Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.

Federation metadata file location:

C:\Users\Administrator\Downloads\spring_saml_metadata.xml

Browse...

☐ Enter data about the relying party manually

Use this option to manually input the necessary data about this relying party organization.

< Previous Next > Cancel

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

Add Relying Party Trust Wizard

Specify Display Name

Enter the display name and any optional notes for this relying party.

Steps

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Display name:

ADFS PCF SSO

Notes:

< Previous Next > Cancel

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

Add Relying Party Trust Wizard

Steps

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?**
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Configure multi-factor authentication settings for this relying party trust. Multi-factor authentication is required if there is a match for any of the specified requirements.

Multi-factor Authentication		Global Settings
Requirements	Users/Groups	Not configured
	Device	Not configured
	Location	Not configured

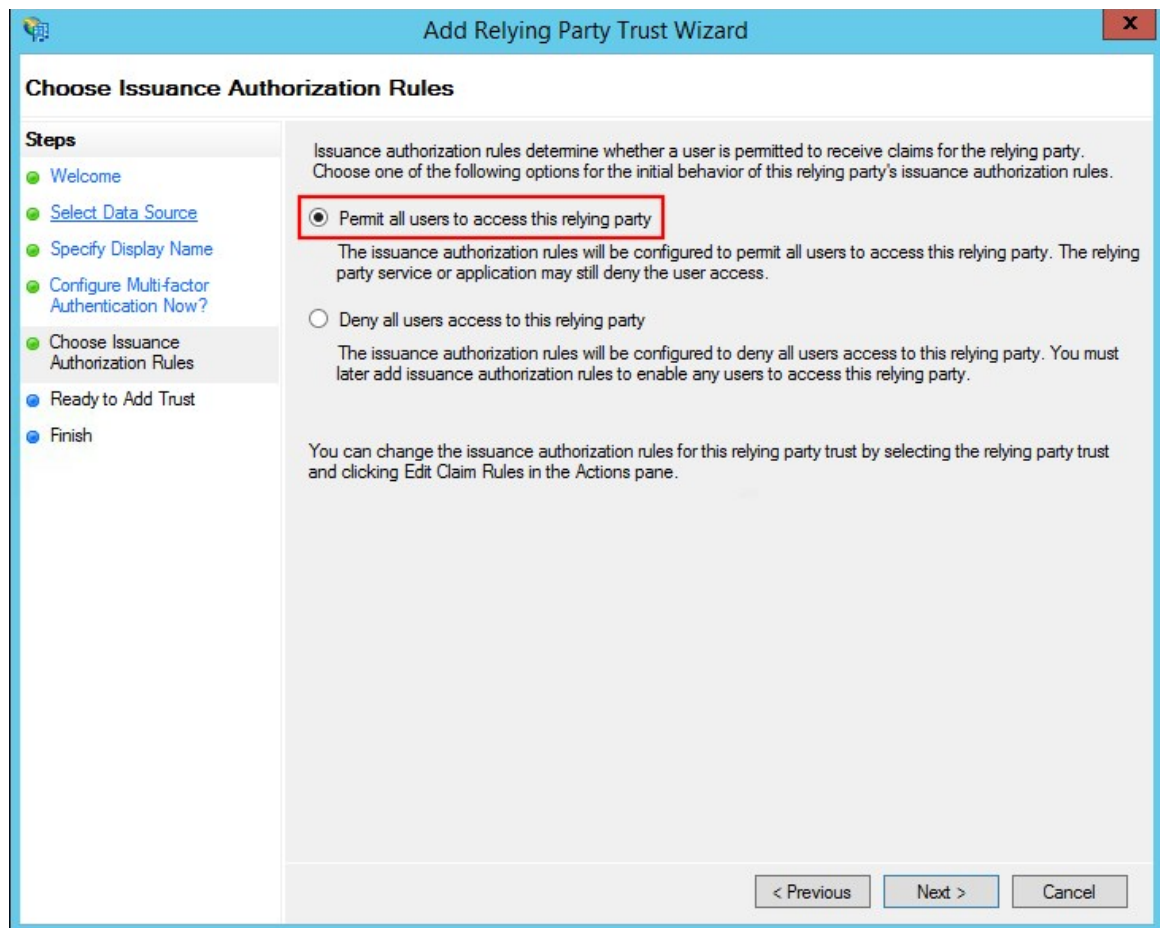
☒ I do not want to configure multi-factor authentication settings for this relying party trust at this time.

☐ Configure multi-factor authentication settings for this relying party trust.

You can also configure multi-factor authentication settings for this relying party trust by navigating to the Authentication Policies node. For more information, see [Configuring Authentication Policies](#).

< Previous Next > Cancel

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:
 1. Click **Add Rule**.
 2. Select **Send LDAP Attributes as Claims** for **Claim rule template** and click **Next**.

The screenshot shows a wizard window titled "Add Transform Claim Rule Wizard". On the left, under "Steps", "Choose Rule Type" is selected with a green dot, and "Configure Claim Rule" is next with a blue dot. The main area has a heading "Select Rule Template" and a sub-instruction: "Select the template for the claim rule that you want to create from the following list. The description provides details about each claim rule template." Below this is a dropdown menu labeled "Claim rule template:" with "Send LDAP Attributes as Claims" selected. This dropdown is highlighted with a red rectangle. Below the dropdown is a text box titled "Claim rule template description:" containing the following text: "Using the Send LDAP Attribute as Claims rule template you can select attributes from an LDAP attribute store such as Active Directory to send as claims to the relying party. Multiple attributes may be sent as multiple claims from a single rule using this rule type. For example, you can use this rule template to create a rule that will extract attribute values for authenticated users from the displayName and telephoneNumber Active Directory attributes and then send those values as two different outgoing claims. This rule may also be used to send all of the user's group memberships. If you want to only send individual group memberships, use the Send Group Membership as a Claim rule template." At the bottom right are three buttons: "< Previous", "Next >", and "Cancel".

3. Enter a **Claim rule name**.
4. Select **Active Directory** for **Attribute store**.
5. Select **E-Mail-Addresses** for **LDAP Attribute** and select **E-mail Address** for **Outgoing Claim Type**.
6. Click **Finish**.

Add Transform Claim Rule Wizard

Configure Rule

Steps

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:
LDAP Email

Rule template: Send LDAP Attributes as Claims

Attribute store:
Active Directory

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	E-Mail-Addresses	E-Mail Address
*		

< Previous Finish Cancel

- Click **Add Rule**.
- Select **Transform an Incoming Claim** for **Claim rule template** and click **Next**.

Add Transform Claim Rule Wizard

Select Rule Template

Steps

- Choose Rule Type
- Configure Claim Rule

Select the template for the claim rule that you want to create from the following list. The description provides details about each claim rule template.

Claim rule template:
Transform an Incoming Claim

Claim rule template description:

Using the Transform an Incoming Claim rule template you can select an incoming claim, change its claim type, and optionally change its claim value. For example, you can use this rule template to create a rule that will send a role claim with the same claim value of an incoming group claim. You can also use this rule to send a group claim with a claim value of "Purchasers" when there is an incoming group claim with a value of "Admins". Multiple claims with the same claim type may be emitted from this rule. Sources of incoming claims vary based on the rules being edited. For more information on the sources of incoming claims, click Help.

< Previous Next > Cancel

- Enter a **Claim rule name**.

10. Select **E-Mail Address** for **Incoming claim type**.
11. Select **Name ID** for **Outgoing claim type**
12. Select **Email** for **Outgoing name ID format**.
13. Click **Finish**.

Add Transform Claim Rule Wizard

Configure Rule

Steps

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name: NameID

Rule template: Transform an Incoming Claim

Incoming claim type: E-Mail Address

Incoming name ID format: Unspecified

Outgoing claim type: Name ID

Outgoing name ID format: Email

☒ Pass through all claim values

☐ Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value: Browse...

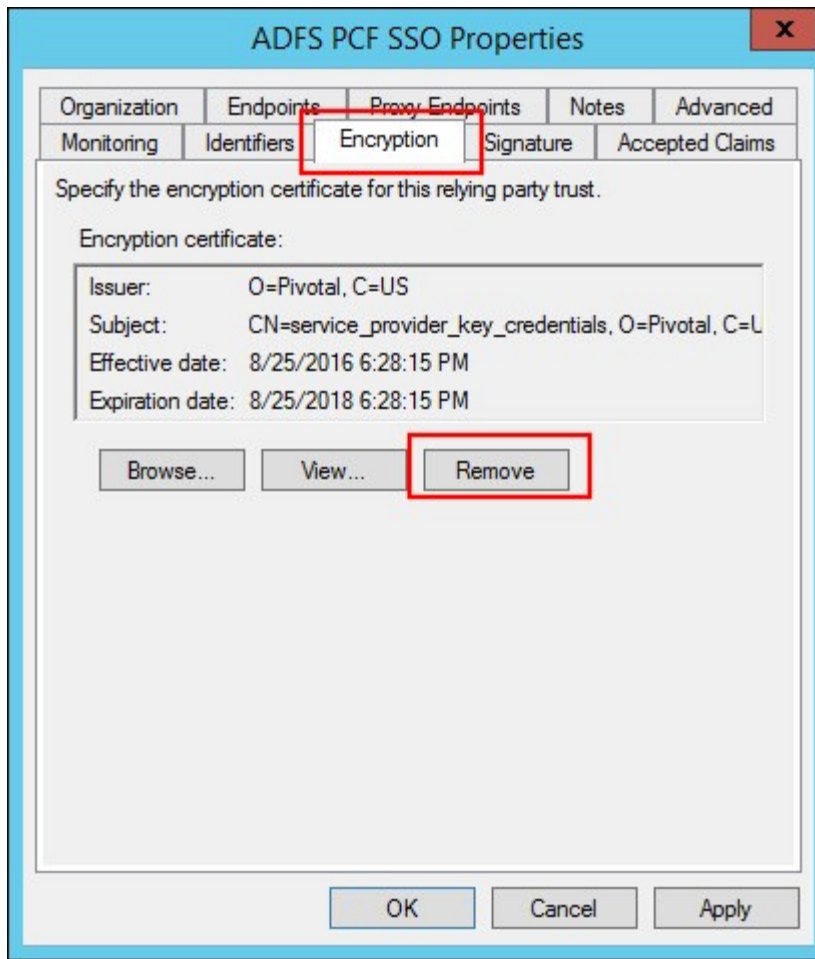
☐ Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

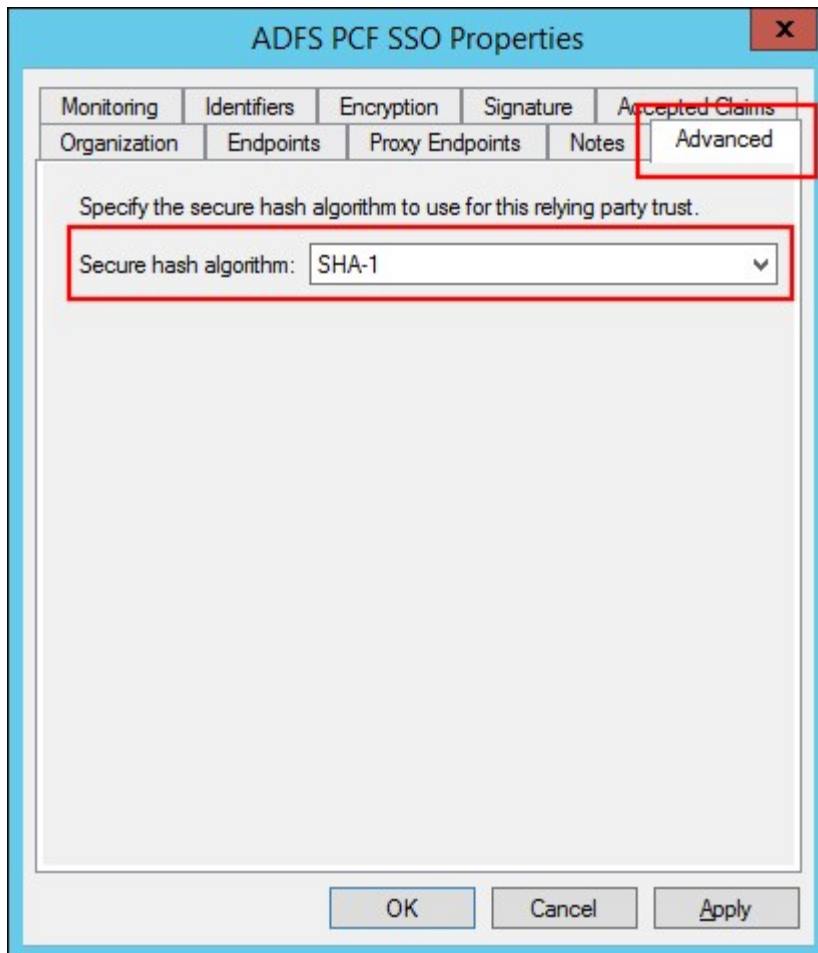
Example: fabrikam.com

< Previous Finish Cancel

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 Pivotal Application Service.



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

1. Open **Windows Powershell** as an Administrator.
2. 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.

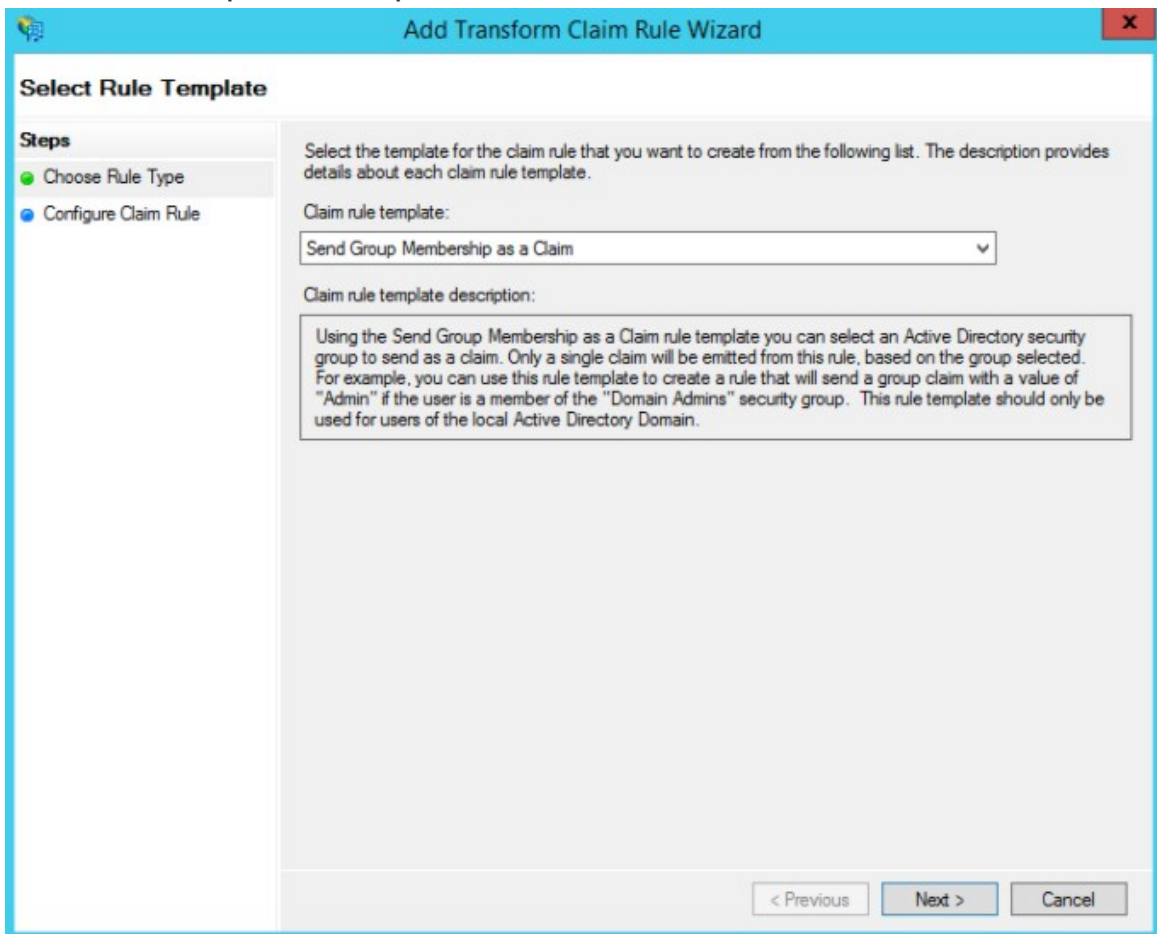
17. (Optional) To specify any application or group attributes that you want to map to users in the ID token, click **Edit Claim Rules...** and configure **Send LDAP Attributes as Claims**. For more information, see the next section.

Setting Up Groups in SAML from ADFS

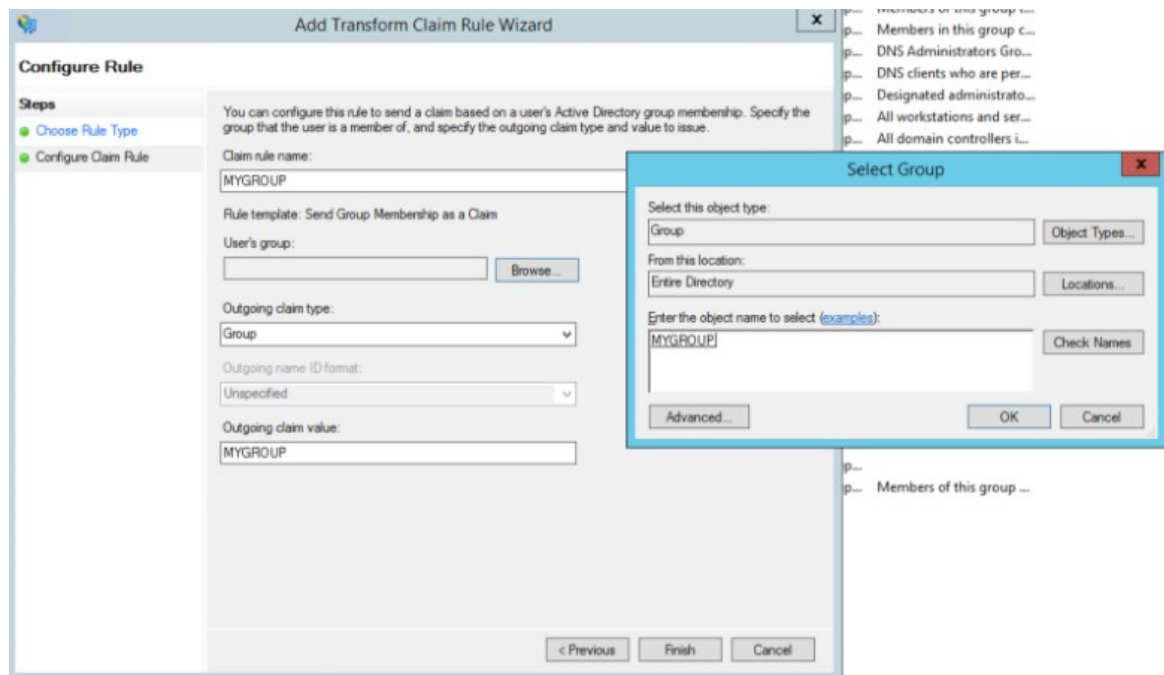
1. Right-click your **Relying Party Trust** and select **Edit Claim Rules...**.



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



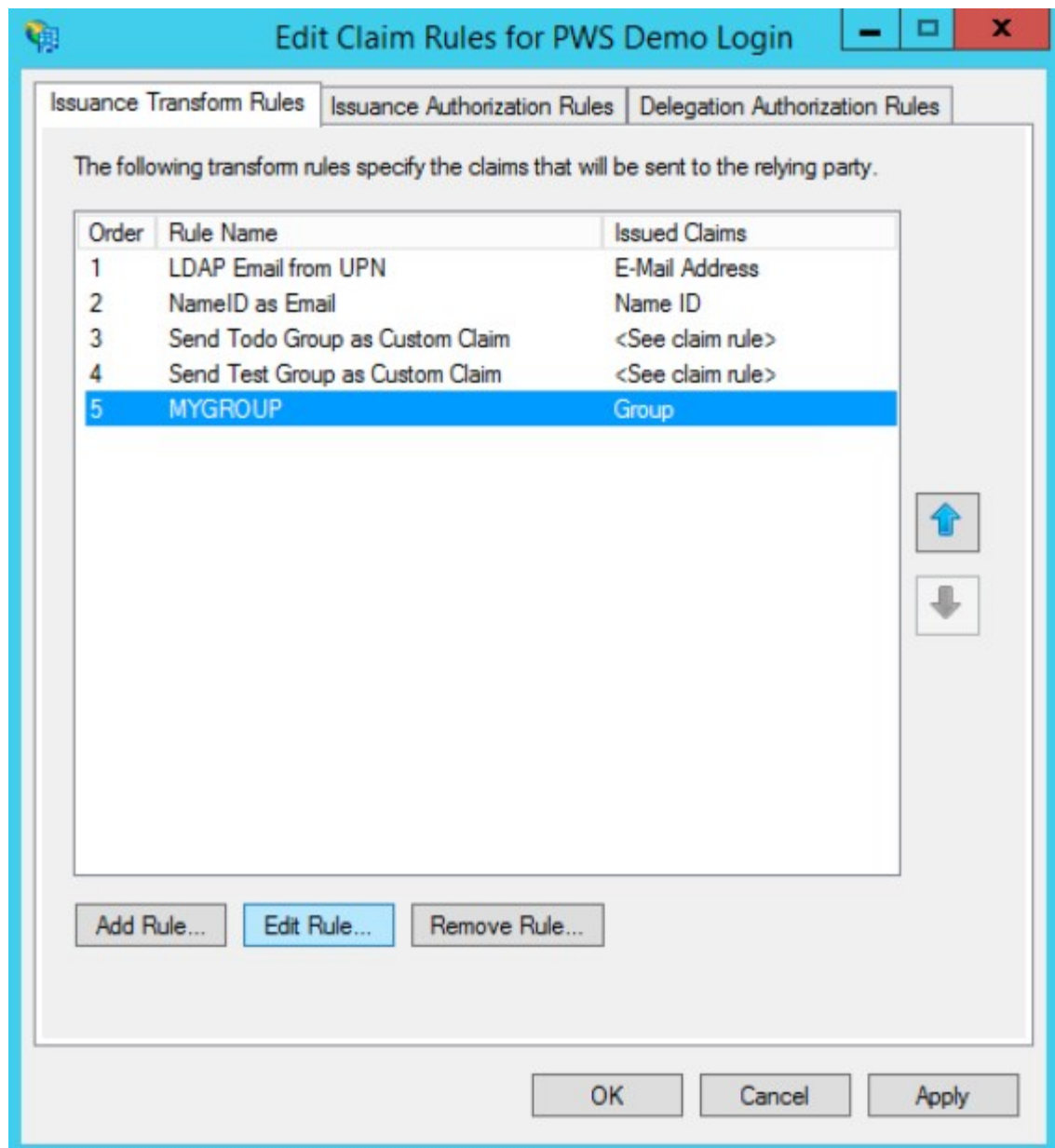
4. Enter the **Claim rule name**.
5. Click **Browse** to select your **User's group**.
6. Select **Group** as your **Outgoing claim type**.
7. Set your **Outgoing claim value** to match your group's name.
8. Click **Finish**.



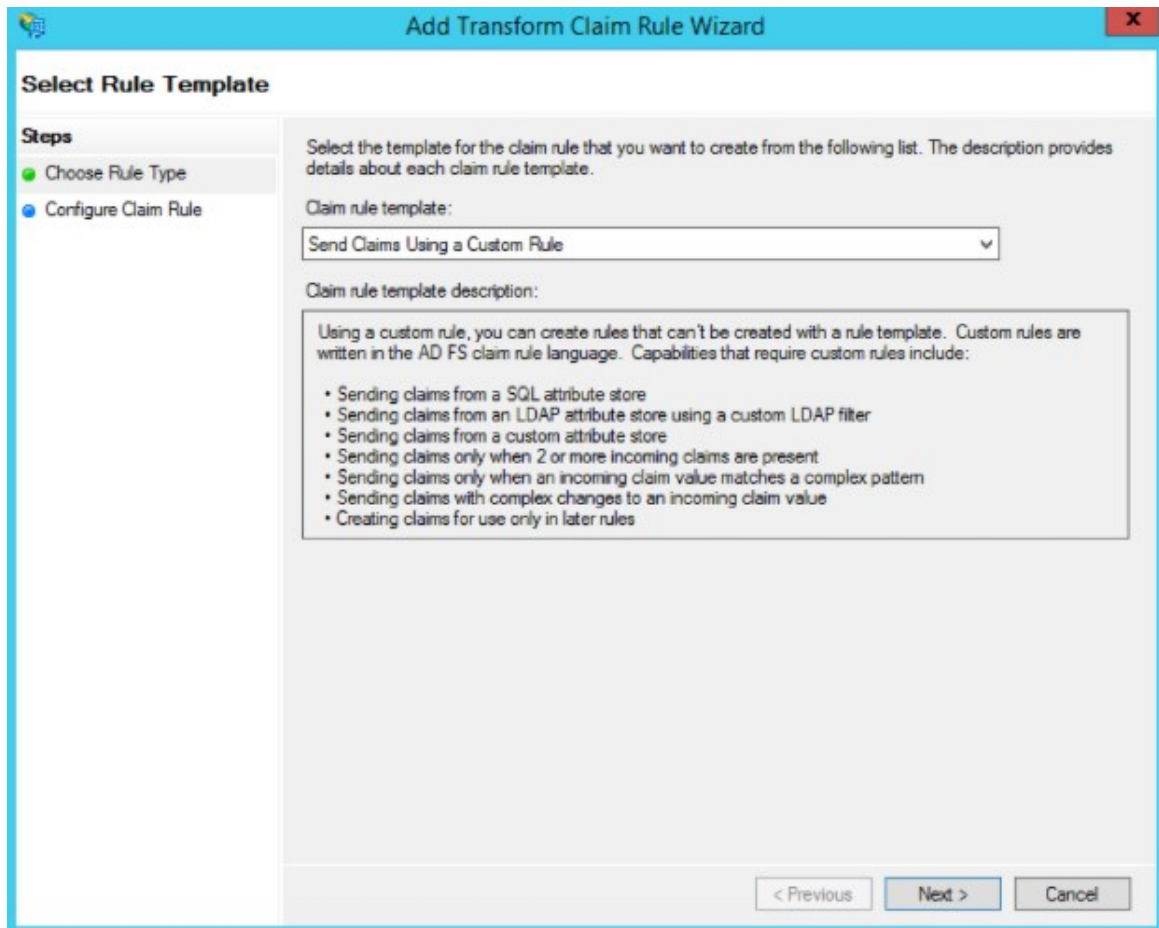
9. To save these configurations and use the default SAML assertion of <http://schemas.xmlsoap.org/claims/Group>, click **OK**. If you want to pass the claims assertion as a custom value “groups” in the SAML assertion, continue to the procedure below.

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".

Edit Rule - MYGROUP

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language.

Claim rule name:

MYGROUP

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value == "S-1-5-21-1881629547-977467862-2541882406-1109", Issuer == "AD
AUTHORITY"]
=> issue(Type = "groups", Value = "MYGROUP", Issuer = c.Issuer,
OriginalIssuer = c.OriginalIssuer, ValueType = c.ValueType);
```

OK Cancel

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

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Configuring a Single Sign-On Service Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

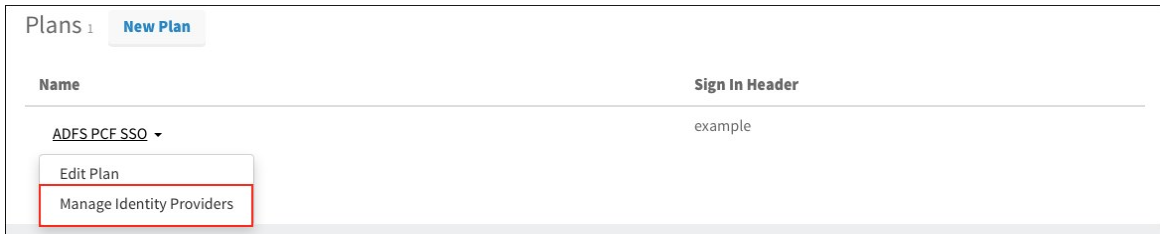
Download Identity Provider Metadata

- Download the metadata from your Active Directory Federation Services server at the

following URL: `https://YOUR-ADFS-HOSTNAME/federationmetadata/2007-06/federationmetadata.xml`

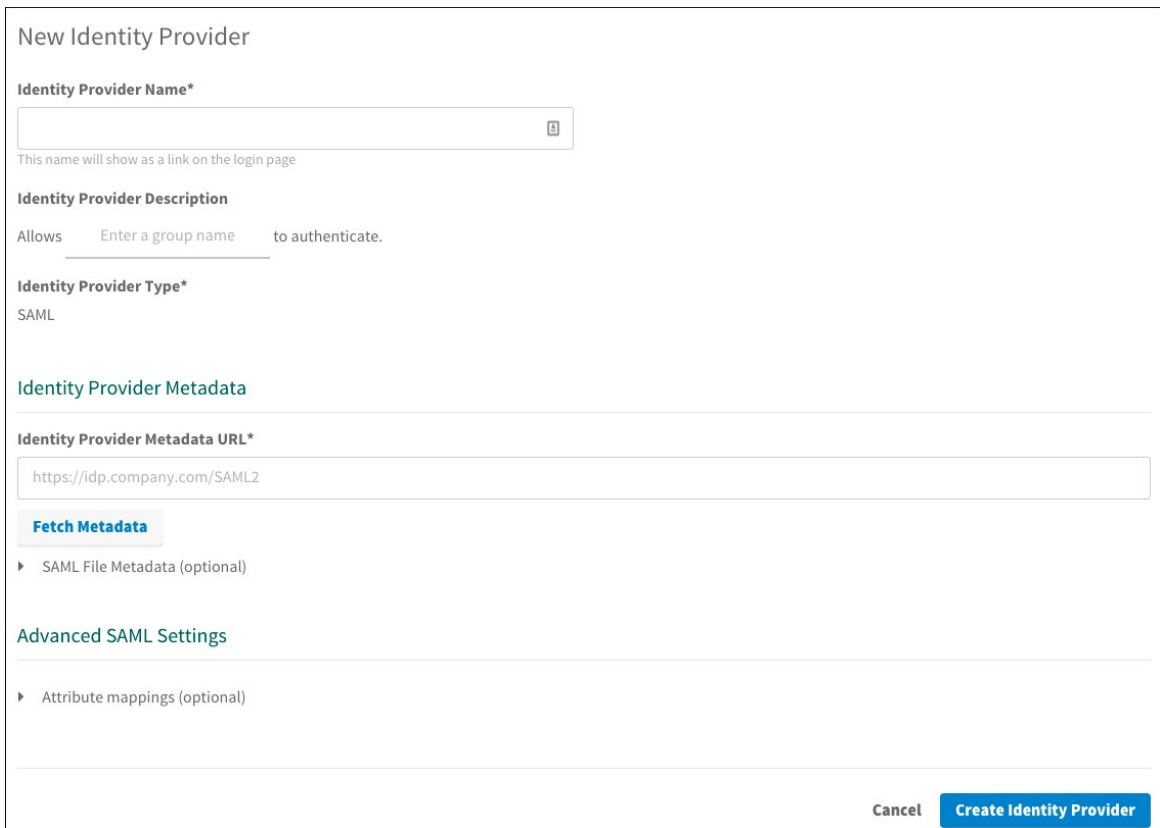
Setting up SAML

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.



The screenshot shows the 'Plans' section of the SSO dashboard. A 'New Plan' button is at the top right. Below it, a table lists plans. The first plan is 'ADFSPCF SSO' with a 'Sign In Header' of 'example'. A dropdown menu is open for the 'ADFSPCF SSO' plan, showing 'Edit Plan' and 'Manage Identity Providers' (which is highlighted with a red box).

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



The screenshot shows the 'New Identity Provider' form. It has the following sections and fields:

- Identity Provider Name***: A text input field with a placeholder icon.
- Identity Provider Description**: A section with the text 'Allows Enter a group name to authenticate.'
- Identity Provider Type***: A dropdown menu with 'SAML' selected.
- Identity Provider Metadata URL***: A text input field containing 'https://idp.company.com/SAML2'.
- Fetch Metadata**: A button.
- Identity Provider Metadata**: A section with a dropdown menu showing 'SAML File Metadata (optional)'.
- Advanced SAML Settings**: A section with a dropdown menu showing 'Attribute mappings (optional)'.
- At the bottom right, there are 'Cancel' and 'Create Identity Provider' buttons.

4. To create a new identity provider, perform the following steps:
 1. Enter an identity provider name in **Identity Provider Name**.
 2. (Optional) Enter a description in **Identity Provider Description**.
 3. Click **SAML File Metadata (optional)**, then click **Upload Identity Provider Metadata** to upload your metadata XML.
 4. (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:
 1. Enter a **Group Name**.
 2. 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:

User Attributes
Map the incoming user attributes to known user schema.

User Schema Attribute	Attribute Name
external_groups	groups

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

User Attributes
Map the incoming user attributes to known user schema.

User Schema Attribute	Attribute Name
external_groups	http://schemas.xmlsoap.org/claims/Group

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 [Create or Edit Resource Permissions](#).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing



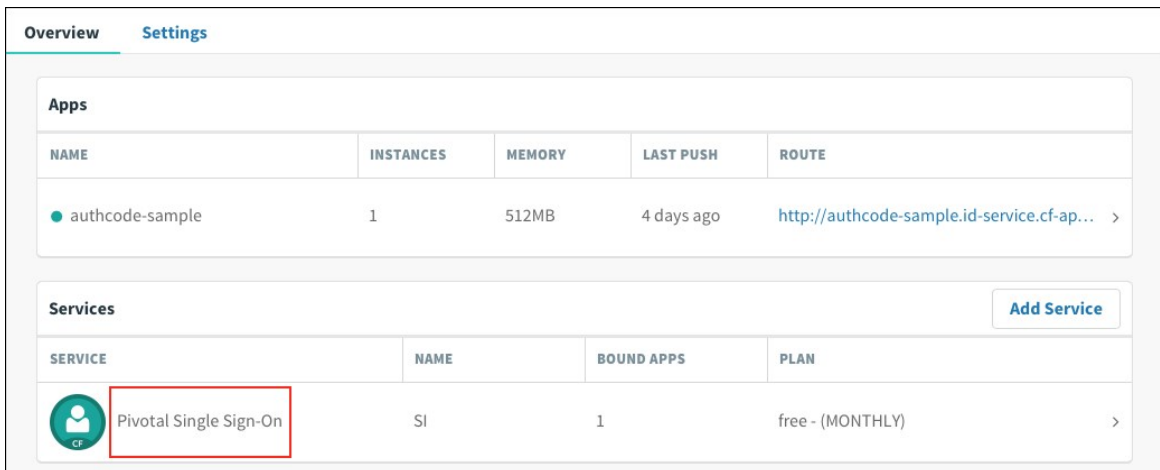
Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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**.



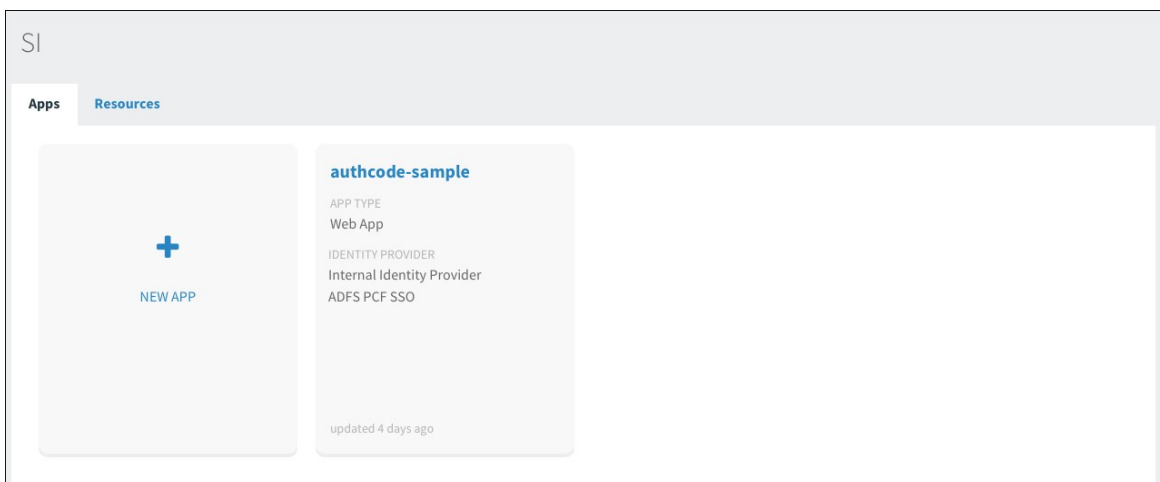
The screenshot shows the 'Services' tab in the Apps Manager interface. It features a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. The 'Pivotal Single Sign-On' service is highlighted with a red box. Below the table, there is a 'Manage' button also highlighted with a red box.

SERVICE	NAME	BOUND APPS	PLAN
 Pivotal Single Sign-On	SI	1	free - (MONTHLY)



The screenshot shows the 'Pivotal Single Sign-On' service instance page. It displays the service name, instance name (SI), and service plan (ADFS PCF SSO). The 'Manage' button is highlighted with a red box.

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



The screenshot shows the 'authcode-sample' application page. It displays the application name, app type (Web App), identity provider (Internal Identity Provider), and service plan (ADFS PCF SSO). The 'authcode-sample' application is highlighted with a red box.

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

authcode-sample Web App Next Steps

App Name*
authcode-sample

Identity Providers

Select a Identity Provider

Internal User Store **ADFS PCF SSO**

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected

Delete Cancel Save Config

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

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap...

- Click the link.

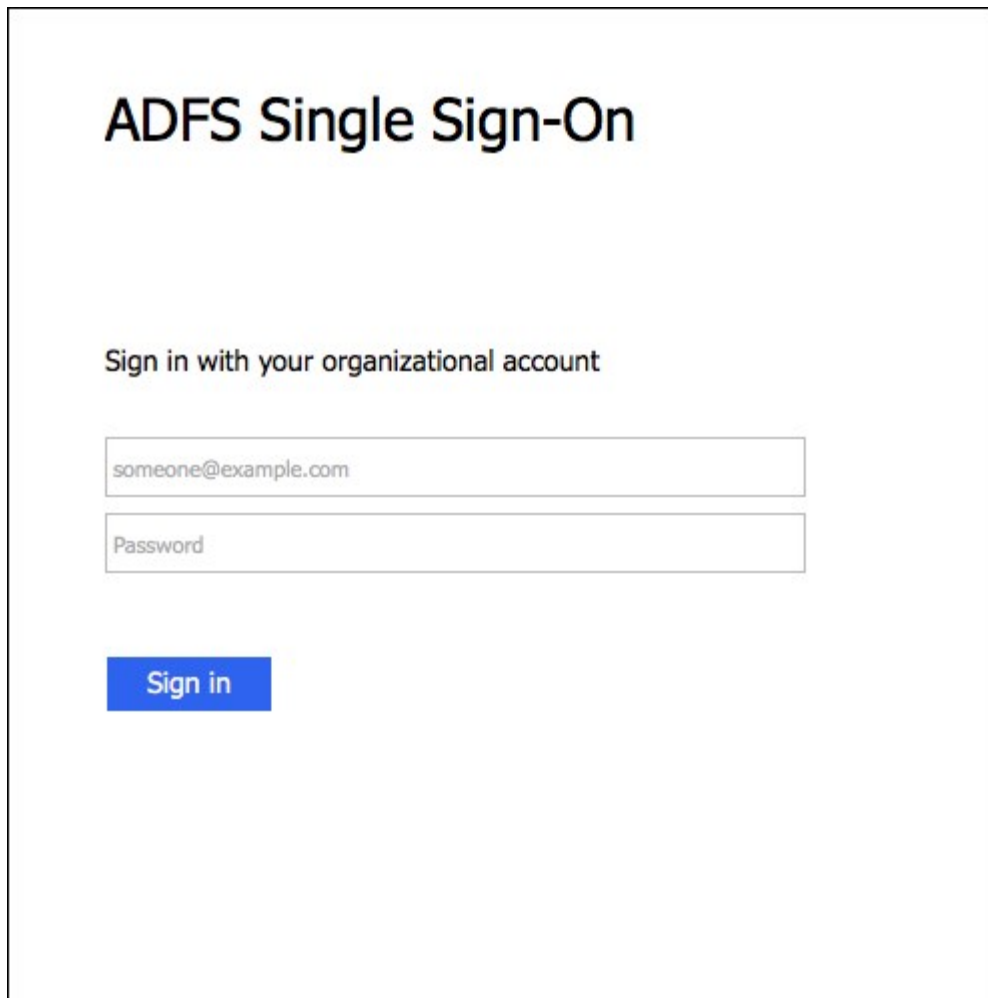
← → ↺ https://authcode-sample

Authcode sample

What do you want to do?

- [Log in via Auth Code Grant Type](#)

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

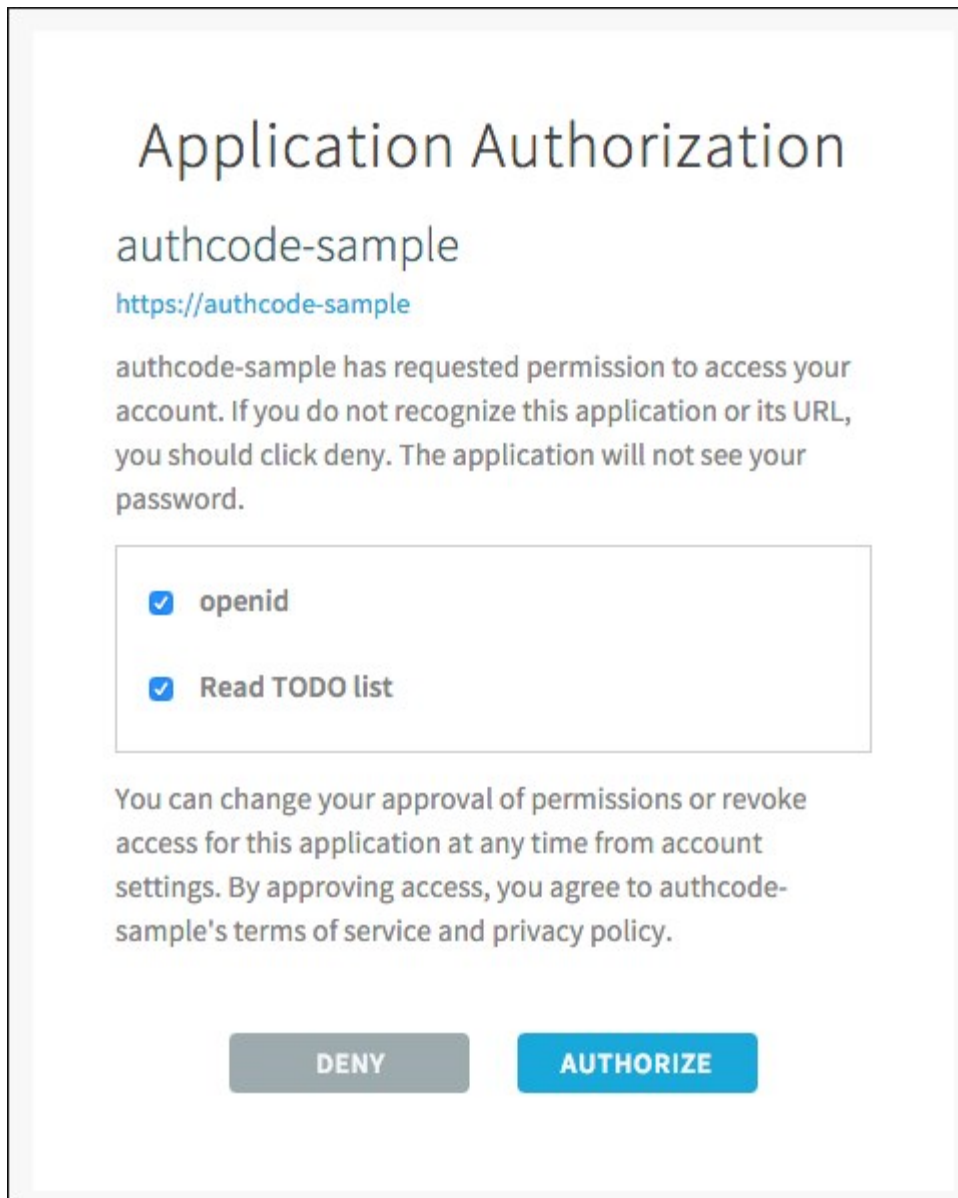


ADFS Single Sign-On

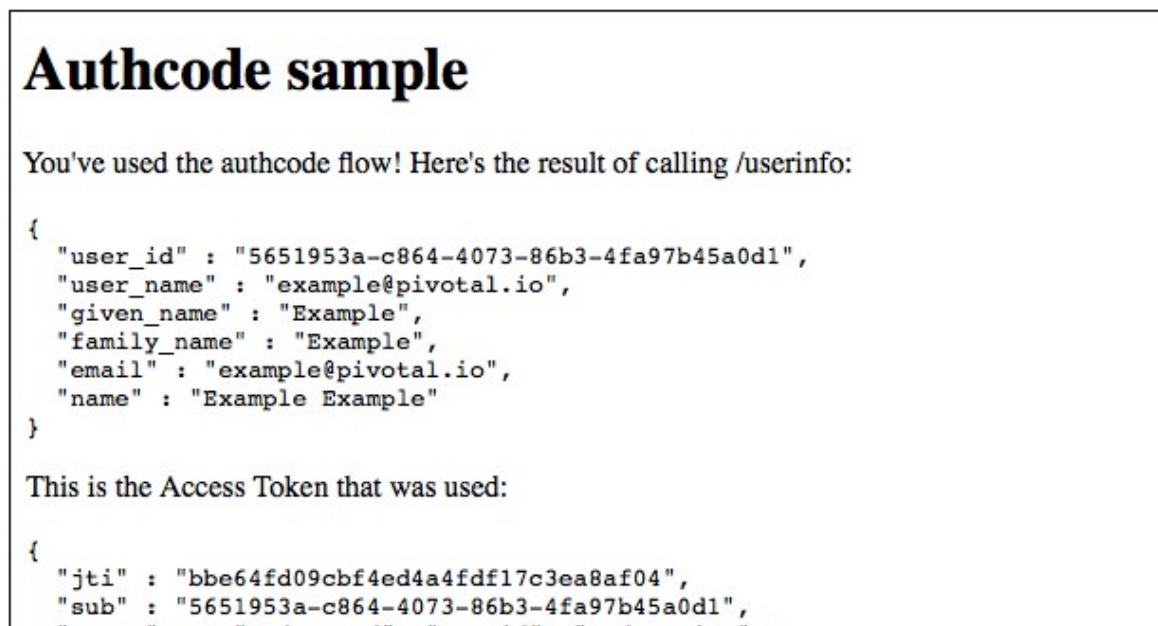
Sign in with your organizational account

Sign in

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



9. The access token and ID token displays.




```

    "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?

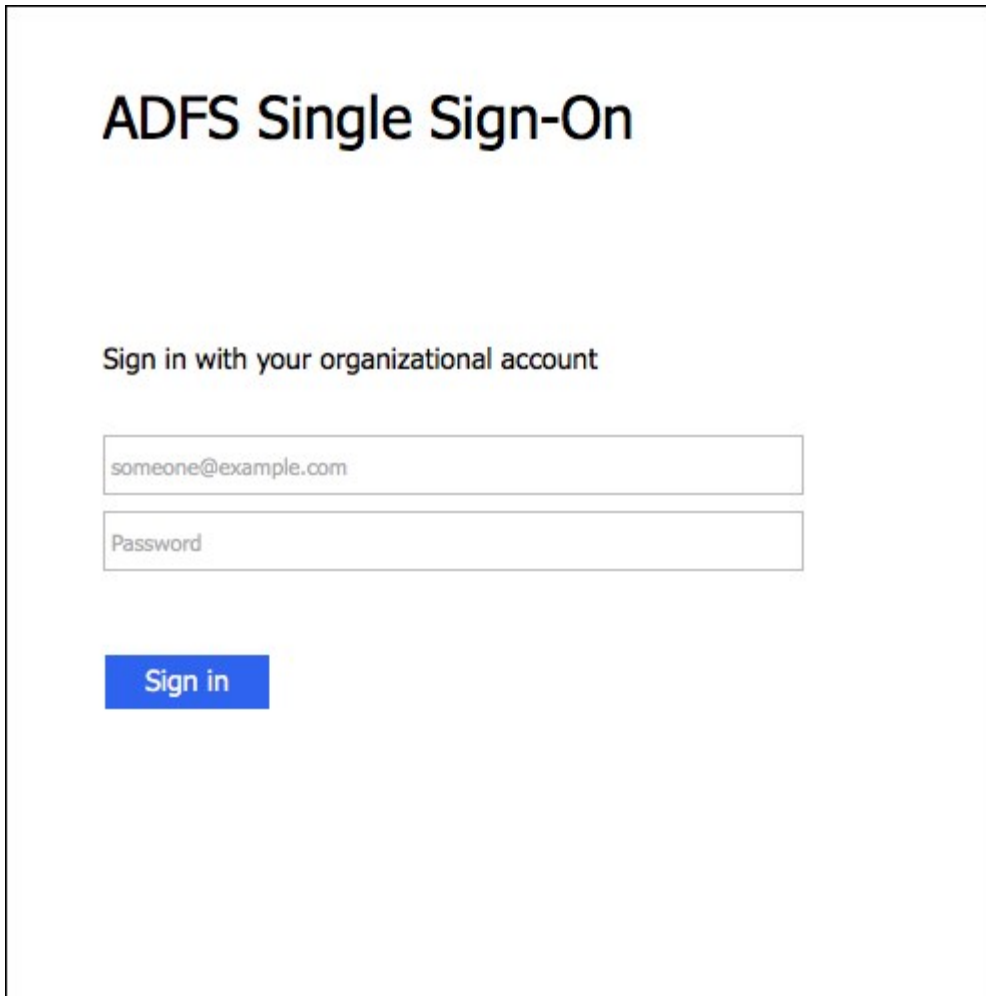
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

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.

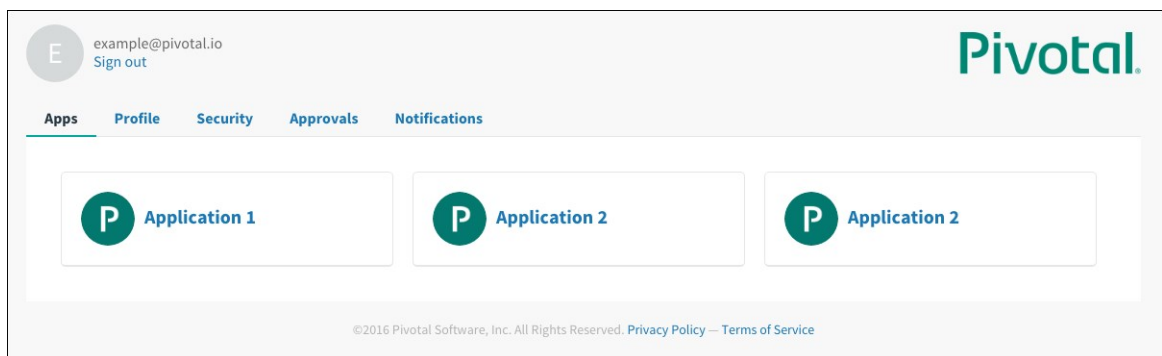
The image shows a web page titled "ADFS Single Sign-On". Below the title, there is a heading "Sign in with your organizational account". Under this heading, there are two input fields: the first contains the email address "someone@example.com" and the second is labeled "Password". Below the input fields is a blue button with the text "Sign in".

ADFS Single Sign-On

Sign in with your organizational account

Sign in

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 ADFS 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**.
-

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](#)

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

ADFS Single Sign-On

Sign in with your organizational account

Sign in

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Troubleshooting

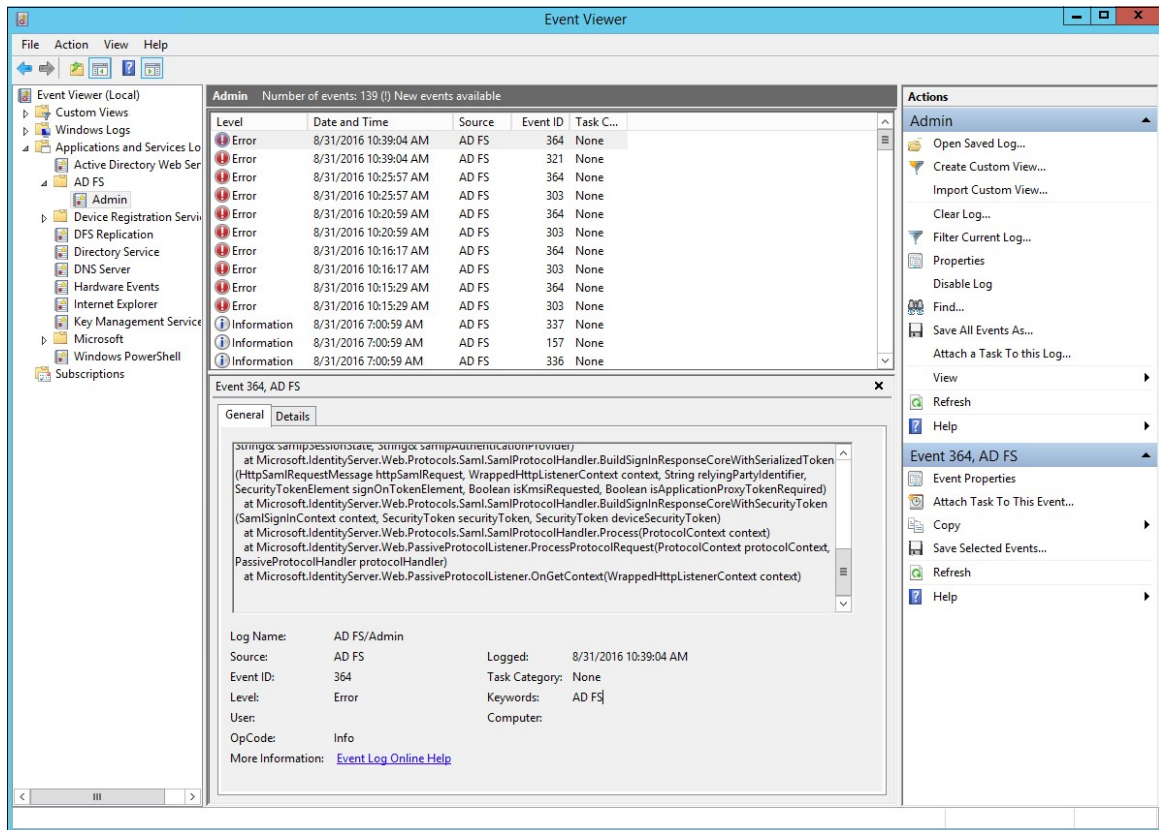


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Azure Active Directory SAML Integration Guide

Azure Active Directory SAML Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

Azure Active Directory

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



Note: To configure SAML, you must have the Pivotal 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Azure Active Directory as a SAML Identity Provider

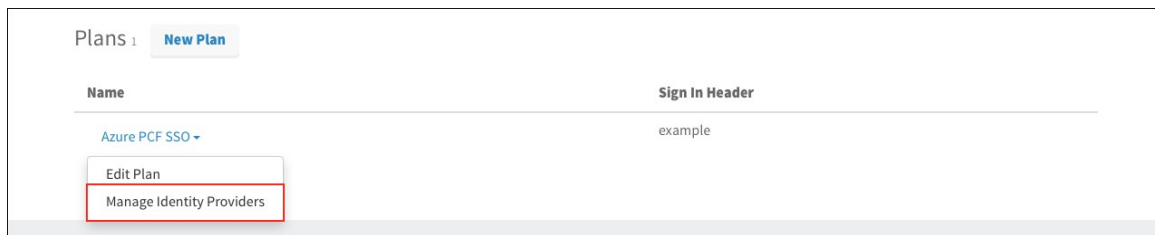


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

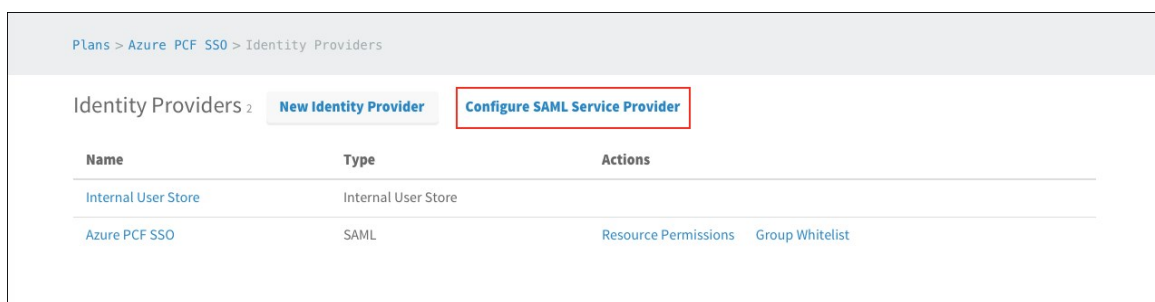
This topic describes how to set up Azure Active Directory (AD) as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry® (PCF) and Azure AD.

Step 1: 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**.



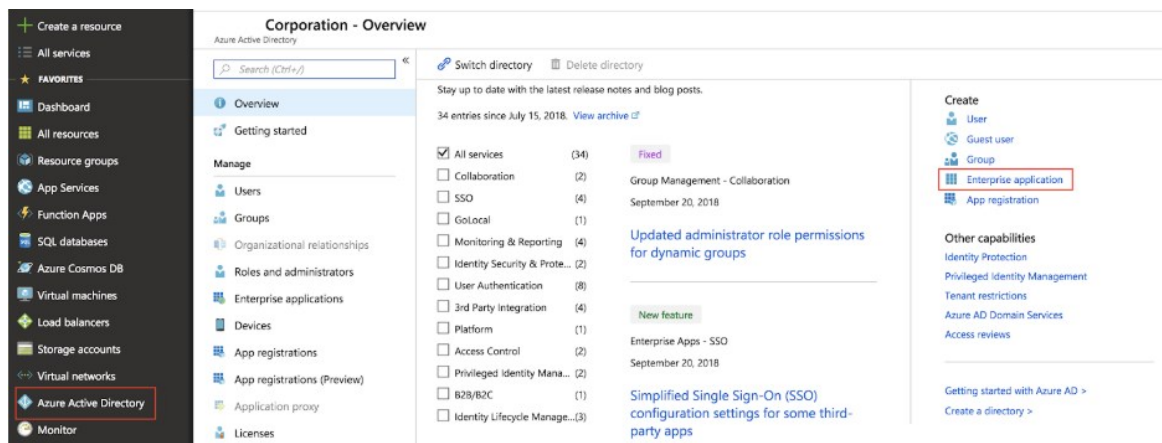
4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.



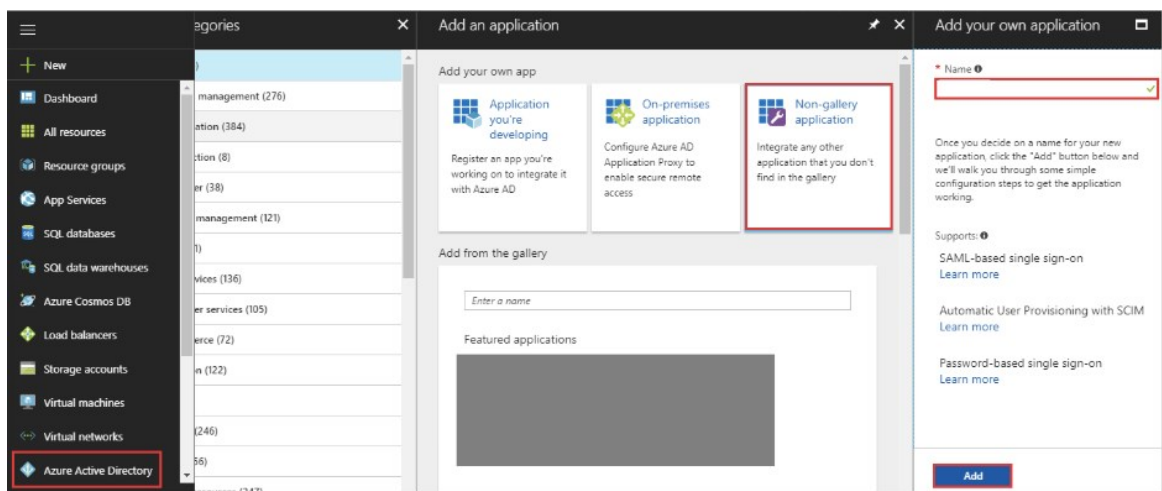
5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

Step 2: Set up SAML in Azure Active Directory (AD)

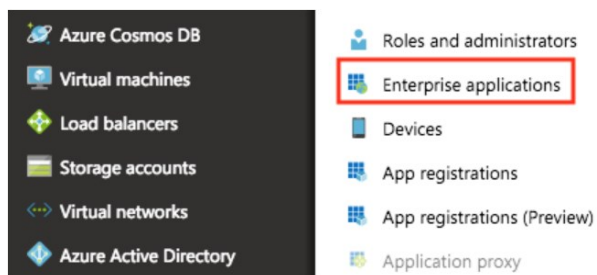
1. Log in to Azure AD as a Global Administrator at <https://portal.azure.com/>.
2. Navigate to **Azure Active Directory** tab > **Enterprise application**.



3. Select **Non-gallery application**. Provide a name and click **Add**.



4. Navigate to **Azure Active Directory > Enterprise applications**.



5. Click your application and then click the **Single sign-on** tab.
6. Select **SAML-based Sign-on** from the dropdown and then click **Upload metadata file** to upload the metadata file you downloaded from step 6 of **Step 1: Set up SAML in PCF**.

7. Record the **App Federation Metadata Url**. You need this for setting up the SSO identity provider configurations. For more information, see [Setting up SAML](#).
8. Provide a **Notification Email** and click **Save**.

4. SAML Signing Certificate [Learn more](#)

Manage the certificate used by Azure AD to sign SAML tokens issued to new_test.

App Federation Metadata Url

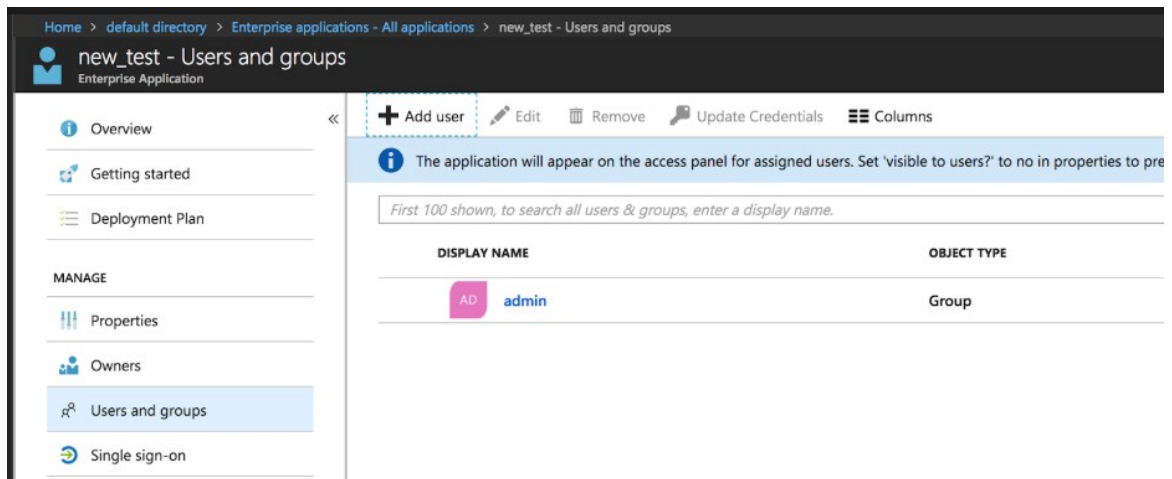
STATUS	EXPIRATION	THUMBPRINT	DOWNLOAD
Active	8/16/2021	1534B2F28AA4563EBD7AA9A15AC23767FD32941D	Certificate (Base64) Certificate (Raw) Metadata XML

[Create new certificate](#)

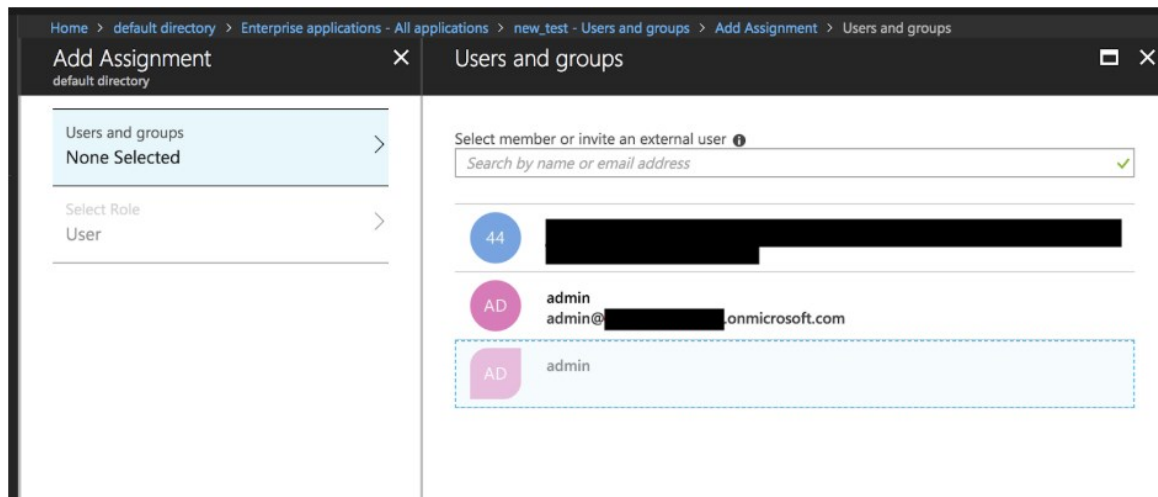
☐ Show advanced certificate signing settings [Learn more](#)

* Notification Email

9. Navigate to **Users and groups** tab and then click **Add User**.

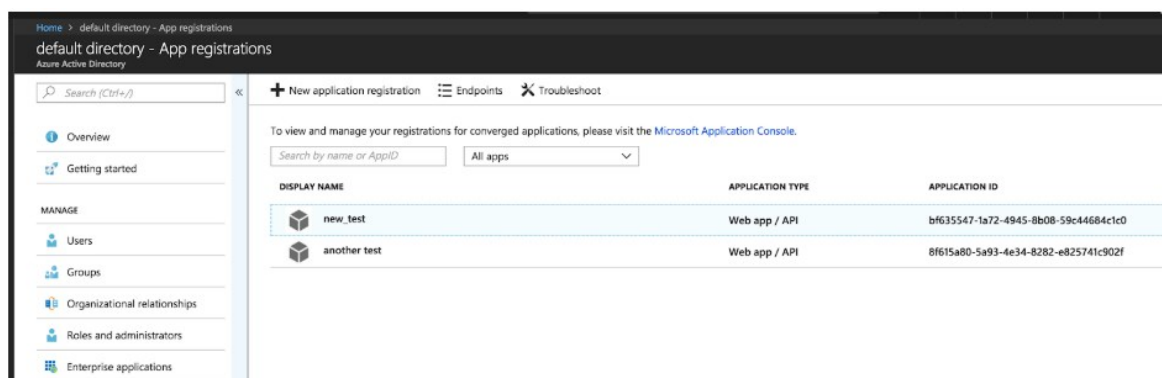


10. Select users or group names from the dropdown. For example, you can add a group that includes all users that should be able to login to the SSO plan.



Step 3: Set up Claims Mapping

1. Navigate to **Azure Active Directory > App registration**. Click your application.



2. To enable user attribute mappings, do the following:
 1. Select the **View and edit all other user attributes** checkbox under the **User Attributes** header.
 2. Modify the attributes.

For more information, see [How to: Customize claims issued in the SAML token for enterprise applications](#).

User Attributes
Edit the user information sent in the SAML token when user sign in to Evernote.

User Identifier:

☒ View and edit all other user attributes

SAML Token Attributes

NAME	VALUE	NAMESPACE
givenname	user.givenname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims ...
surname	user.surname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims ...
emailaddress	user.mail	http://schemas.xmlsoap.org/ws/2005/05/identity/claims ...
name	user.useridprincipalname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims ...

[Add attribute](#)

3. To pass group membership claims to the application, do the following:
 1. Click **Manifest**.
 2. Locate `groupMembershipClaims` and set the value to one of the following:
 - `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.
 3. Save the change.

For more information, see [How to: Customize claims issued in the SAML token for enterprise applications](#).

Edit manifest

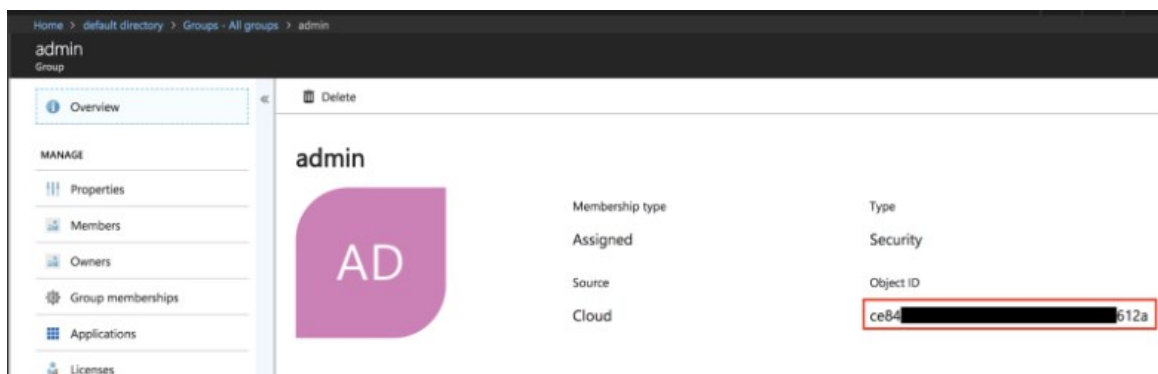
Save Discard Edit Upload Download

```

1  [
2    "appId": "[REDACTED]",
3    "appRoles": [
4      {
5        "allowedMemberTypes": [
6          "User"
7        ],
8        "displayName": "User",
9        "id": "[REDACTED]",
10       "isEnabled": true,
11       "description": "User",
12       "value": null
13     },
14     {
15       "allowedMemberTypes": [
16         "User"
17       ],
18       "displayName": "msiam_access",
19       "id": "[REDACTED]",
20       "isEnabled": true,
21       "description": "msiam_access",
22       "value": null
23     }
24   ],
25   "availableToOtherTenants": false,
26   "displayName": "new_test",
27   "errorUrl": null,
28   "groupMembershipClaims": "SecurityGroup",
29 ]

```

4. Navigate to **Azure Active Directory > Groups**.
5. For each group that is used by the SSO plan, record the **Object ID**. Azure AD will pass the Object ID of these groups to the SSO plan. For more information, see [Configure Group Permissions](#).



Create a pull request or raise an issue on the source for this page in GitHub

Configuring a Single Sign-On Service Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Step 1: 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.

Name	Sign In Header
Azure PCF SSO	example

Buttons: Edit Plan, **Manage Identity Providers**

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

New Identity Provider

Identity Provider Name*

This name will show as a link on the login page

Identity Provider Description

Allows Enter a group name to authenticate.

Identity Provider Type*

SAML

Identity Provider Metadata

Identity Provider Metadata URL*

[https://idp.company.com/SAML2](#)

Fetch Metadata

▶ SAML File Metadata (optional)

Advanced SAML Settings

▶ Attribute mappings (optional)

Cancel **Create Identity Provider**

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

1. Enter an **Identity Provider Name**.
2. (Optional) Enter an **Identity Provider Description**.
3. Enter the **App Federation Metadata Url** you obtained from step 7 in [Step 2: Set up SAML in Azure Active Directory \(AD\)](#) and click **Fetch Metadata**.
4. (Optional) Enter mappings under **Advanced SAML Settings > Attribute Mappings**.
5. Click **Create Identity Provider**.

Step 2: Configure Group Permissions



Note: Azure AD will pass the Object ID of the groups recorded in step 5 of [Step 3: Set up Claims Mapping](#) to the SSO plan.

1. Add groups to be propagated from the external identity provider to the ID token by following these steps:
 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 **Group Whitelist** next to your identity provider.
 4. Enter the group names.
 5. Click **Save Group Whitelist**.
2. Map the groups to resources defined in the SSO service by following these steps:
 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 **Resource Permissions**.
 4. Click **New Permissions Mapping** and perform the following steps:
 1. Enter a **Group Name**.
 2. For **Select Permissions**, select the permissions that the members of the group from the external identity provider should have access to.
 3. Click **Save Permissions Mapping**.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Test Your Configurations in Azure AD

1. Log in to Azure AD at <https://portal.azure.com/>.
2. Navigate to **Azure Active Directory** > **Enterprise Applications**.
3. Select your application and navigate to **Single Sign-on** > **Test SAML settings**.
4. Select the user that you want to log in as.

If you have setup all configuration correctly, you should see something like the images below. Otherwise, you should see some meaningful error message.

Test single sign-on with new_test

Please make sure you have configured new_test before testing.

Sign in as current user

Sign in as someone else

✔ Azure AD successfully issued a token (SAML response) to the application (service provider). If you still can't access the application you need to contact the software vendor and share the information below.

- [Download the SAML request](#)
- [Download the SAML response](#)

✓ User unique identifier (NameID)

✓ Token Claims

✓ Token signing certificate


For more information see: [I can complete Azure AD sign in, but I'm seeing an error on the application's sign in page](#) [↗](#)

^ Token Claims	
NAME	VALUE
http://schemas.microsoft.com/identity/claims/tenantid	31854a85-5cdb-4219-8e...
http://schemas.microsoft.com/identity/claims/objectidentifier	4f032547-e839-4579-8e0...
http://schemas.microsoft.com/identity/claims/displayname	42fdf263-ac43-4f74-8754...
http://schemas.microsoft.com/ws/2008/06/identity/claims/groups	981769e2-5962-4646-b2...


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**.

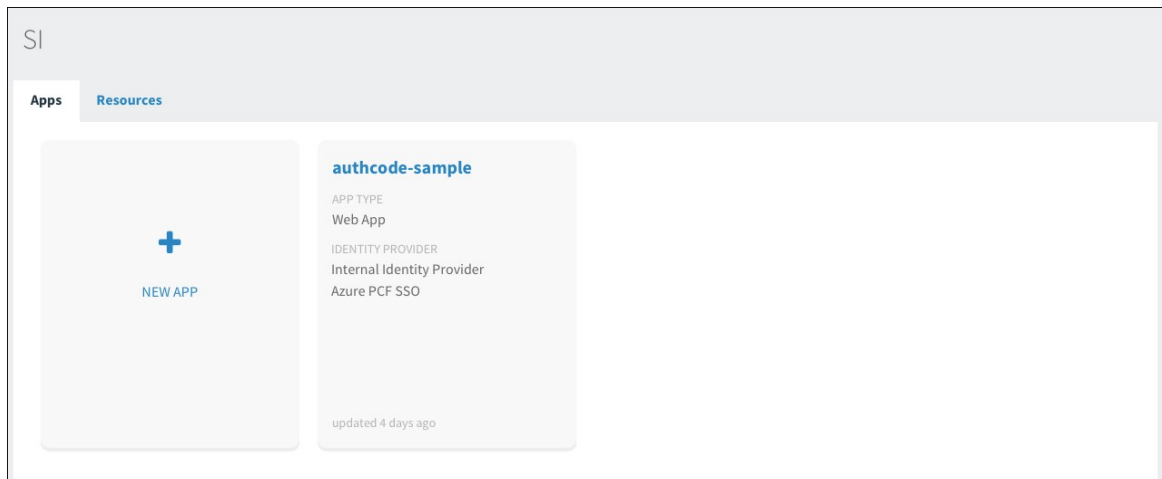
The screenshot shows the 'Services' tab in the Apps Manager interface. It features a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. The 'Pivotal Single Sign-On' service is listed with instance name 'SI' and plan 'free - (MONTHLY)'. A red box highlights the service name and its icon.

SERVICE	NAME	BOUND APPS	PLAN
 Pivotal Single Sign-On	SI	1	free - (MONTHLY)

The screenshot shows the details page for the 'Pivotal Single Sign-On' service. It includes tabs for 'App Binding (1)', 'Plan', and 'Settings'. The 'Manage' button is highlighted with a red box. Below the tabs, there is a 'Bound Apps' section showing the application 'authcode-sample'.

SERVICE	INSTANCE NAME	SERVICE PLAN
 Pivotal Single Sign-On	SI	Azure PCF SSO

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



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

authcode-sample Web App Next Steps

App Name*

authcode-sample

Identity Providers

Select a Identity Provider

Internal User Store **Azure PCF SSO**

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected ▾

Delete Cancel Save Config

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

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap... >

6. Click the link.



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

Microsoft Azure

Work or school, or personal Microsoft account

Email or phone

Password

☐ Keep me signed in

Sign in **Back**

[Can't access your account?](#)

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

Application Authorization

authcode-sample

<https://authcode-sample>

authcode-sample has requested permission to access your account. If you do not recognize this application or its URL, you should click deny. The application will not see your password.

☒ openid

☒ Read TODO list

You can change your approval of permissions or revoke access for this application at any time from account settings. By approving access, you agree to authcode-sample's terms of service and privacy policy.

DENY
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?

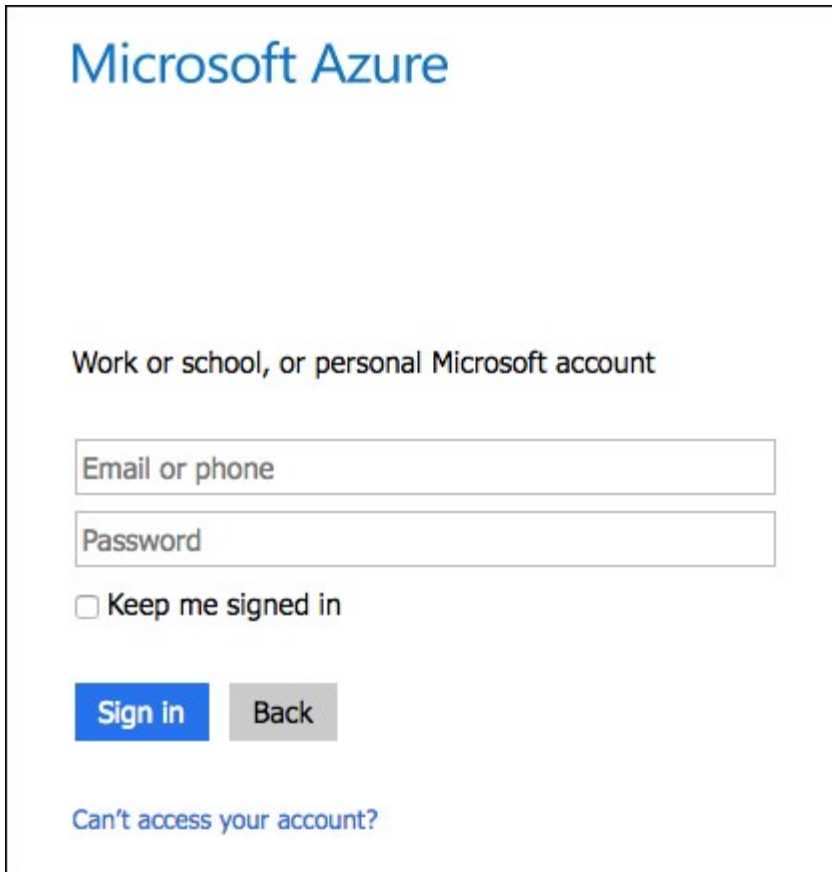
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

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.

The image shows the Microsoft Azure login interface. At the top is the "Microsoft Azure" logo. Below it is the text "Work or school, or personal Microsoft account". There are two input fields: "Email or phone" and "Password". Below the password field is a checkbox labeled "Keep me signed in". There are two buttons: a blue "Sign in" button and a grey "Back" button. At the bottom is a link that says "Can't access your account?".

Microsoft Azure

Work or school, or personal Microsoft account

Email or phone

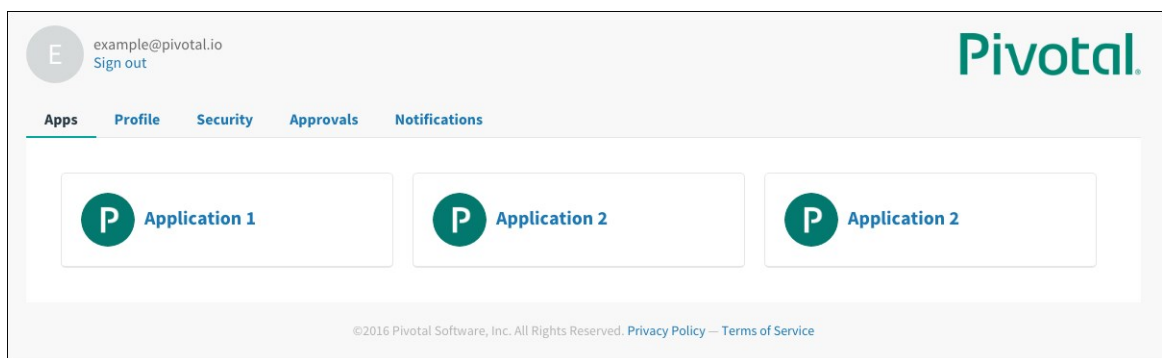
Password

☐ Keep me signed in

Sign in Back

[Can't access your account?](#)

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

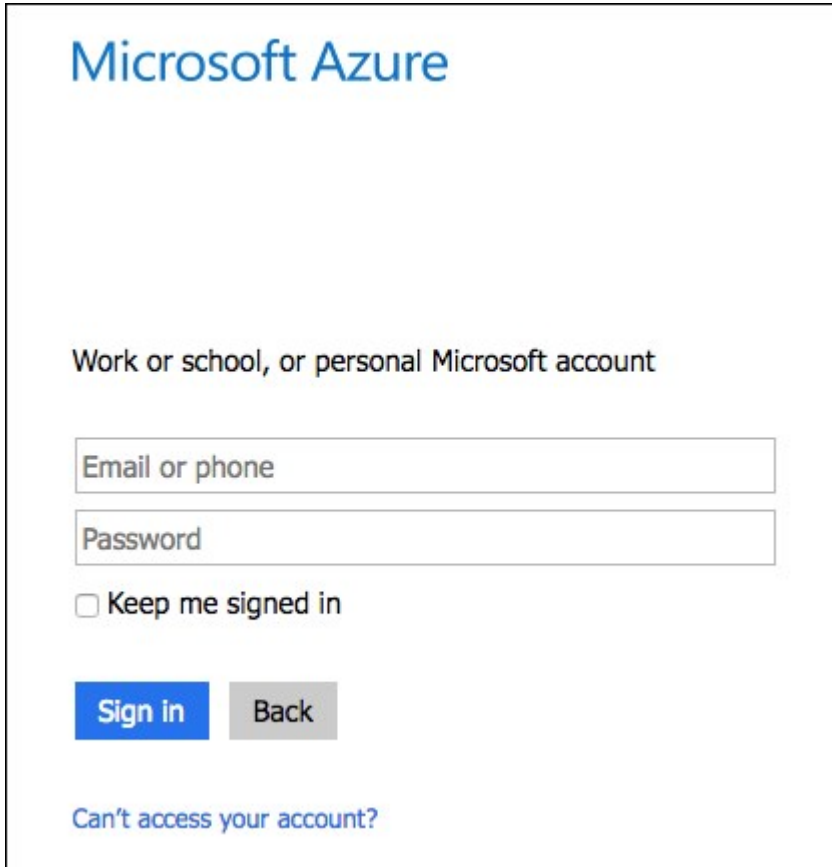
Test single sign-off to ensure that when users log out of the application, they are logged out of Azure AD as well.

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**.
-

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](#)

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



[Create a pull request or raise an issue on the source for this page in GitHub](#)

Troubleshooting



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Failed Login

Symptom:

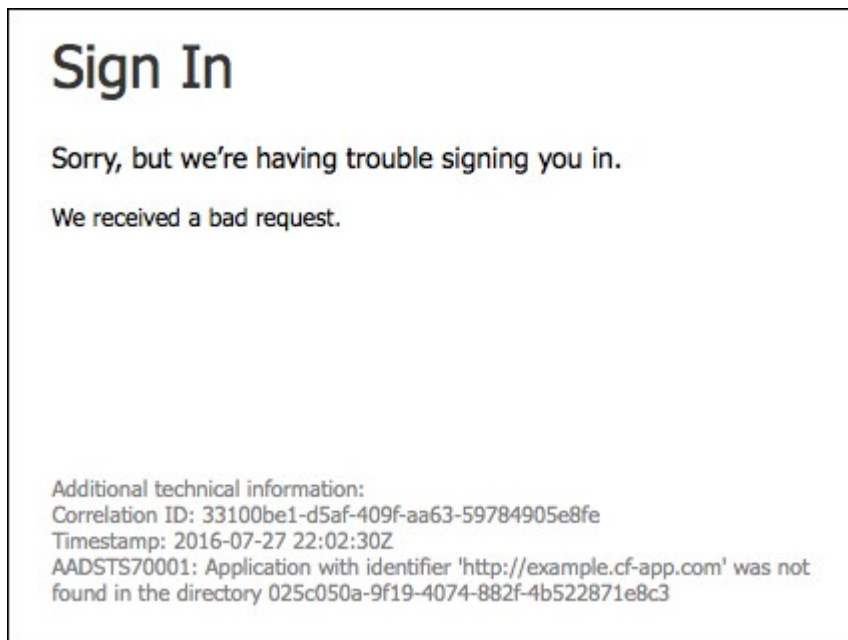
You cannot log in to your SSO plan.

Solutions:

- Pivotal recommends using a different browser or deleting your browser cache and history before you log in to your SSO plan. Your SSO plan can fail if you are already logged in to Azure AD as the Global Administrator account that was used to set up all the configurations.
- If your login fails more than five times, Azure locks your account for 30 minutes. There is currently no way to unlock an account in Azure AD, so wait for the lockout period.
- Pivotal recommends testing your SSO plan from Azure AD to see the contents of the SAML assertion. For more information, see [Test Your Configurations in Azure AD](#).

App ID Not Found

Symptom:



Explanations:

- The App ID URI is misconfigured on Azure AD.

Reply URL Does Not Match

Symptom:

Sign In

Sorry, but we're having trouble signing you in.

We received a bad request.

Additional technical information:
Correlation ID: 148c57c2-6082-493c-9dd9-2c646bf0f0b9
Timestamp: 2016-07-27 22:03:47Z
AADSTS50011: The reply address 'https://example.cf-app.com/saml/SSO/alias/example.cf-app.com' does not match the reply addresses configured for the application: http://example.cf-app.com.

Explanation:

- The Reply URL is misconfigured on Azure AD.

Missing Name ID

Symptom:

Identity Provider Metadata

Identity Provider Metadata URL *

[Fetch Metadata](#)

Error processing metadata

▼ SAML File Metadata (optional)

[Upload Identity Provider Metadata](#) federationmetadata.xml


Explanation:

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

Create a [pull request](#) or raise an [issue](#) on the source for this page in GitHub

Azure Active Directory OIDC Integration Guide

Azure Active Directory OIDC Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support](#)

Lifecycle Policy. To stay up to date with the latest software and security updates, upgrade to a supported version.

This documentation introduces how to set up Azure Active Directory (Azure AD) with Open ID Connect (OIDC) 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.

To set up Azure AD with Security Assertion Markup Language (SAML), see the [Azure Active Directory SAML Integration Guide](#).

Prerequisites

To integrate Azure AD with Single Sign-On service using OIDC, you need the following:

Pivotal

- PCF, v1.12 or later.
- Single Sign-On, v1.5.0 or later.

Azure AD

- An active Azure AD tenant.
- A user with admin privileges.



Note: To configure OIDC, you must have the Pivotal 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 the following steps to integrate your deployment with Azure AD and SSO.

1. [Configure Azure AD as an OIDC Identity Provider](#)
2. [Configure a Single Sign-On Service Provider](#)

Testing and Troubleshooting

- [Testing](#)
- [Troubleshooting](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Azure Active Directory as an OIDC Identity Provider

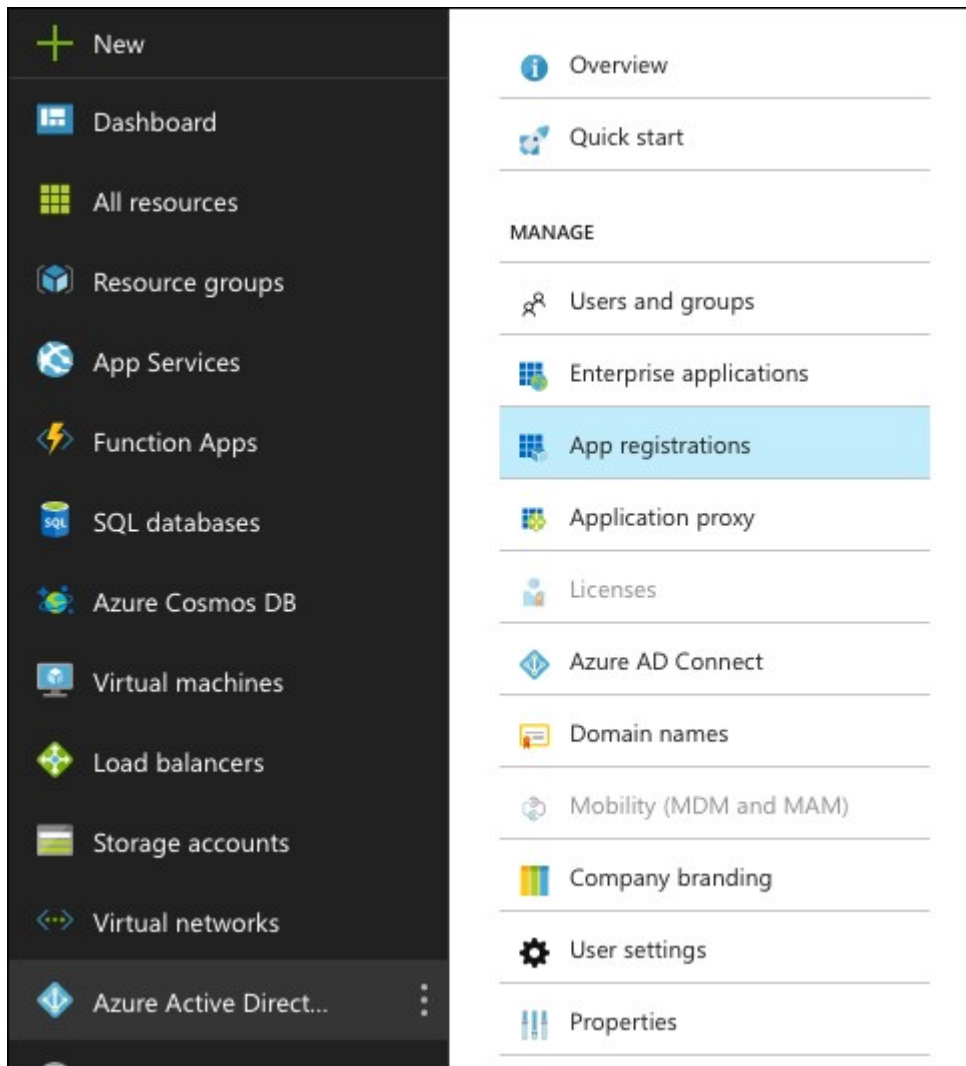


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

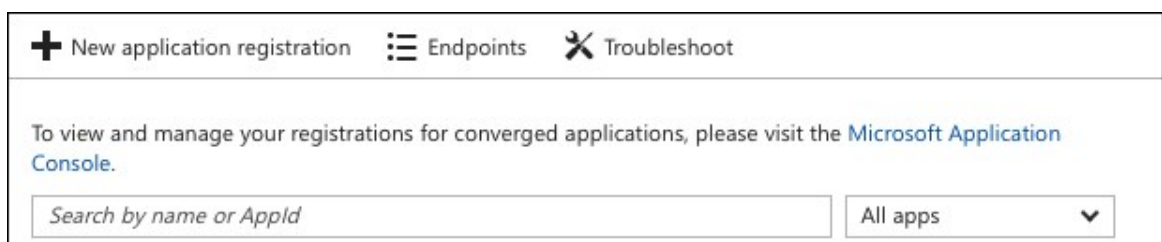
This topic describes how to integrate Azure Active Directory (Azure AD) as an identity provider for a Single Sign-On (SSO) service plan, by configuring OpenID Connect (OIDC) in both Pivotal Cloud Foundry (PCF) and Azure AD.

Follow the steps below to set up relying party in Azure AD.

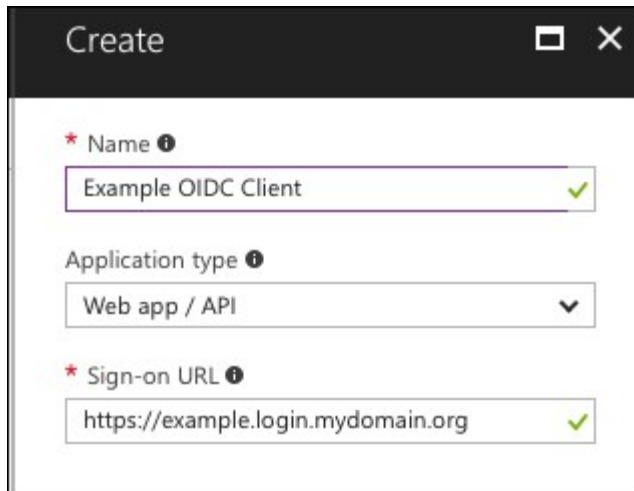
1. Log in to your Azure account and navigate to **Azure Active Directory** > **App registrations**.



2. Select **+** to create a **New application registration**. A configuration pane appears.



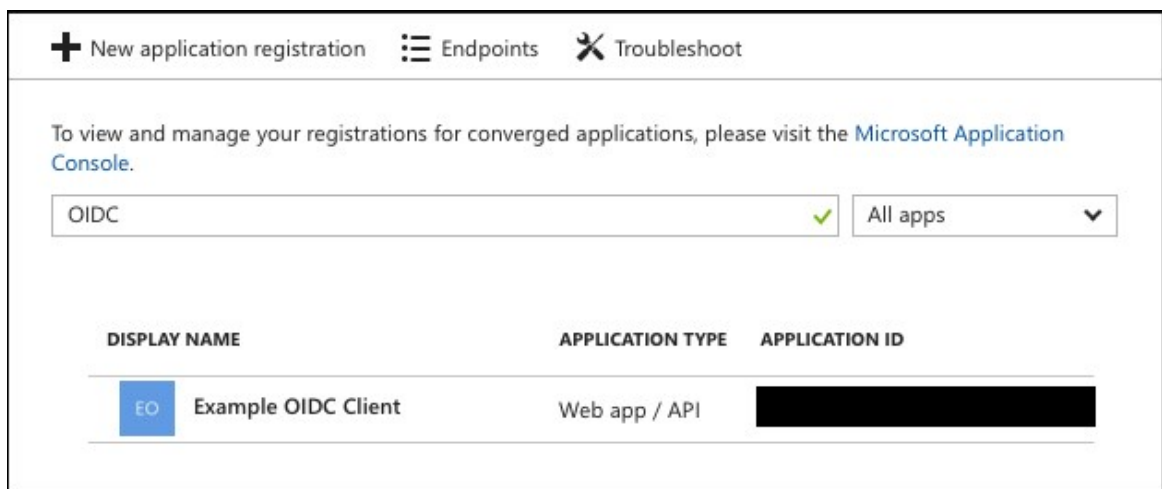
- Under Application type, select **Web App/API** and enter any Name and any Sign-on URI. You can optionally enter the full **Auth Domain** URL generated based on the **Auth Domain** setting you used when you created the service plan that you are integrating with Azure AD.




The 'Create' form contains the following fields:

- Name**: Example OIDC Client (with a green checkmark)
- Application type**: Web app / API (dropdown menu)
- Sign-on URL**: https://example.login.mydomain.org (with a green checkmark)

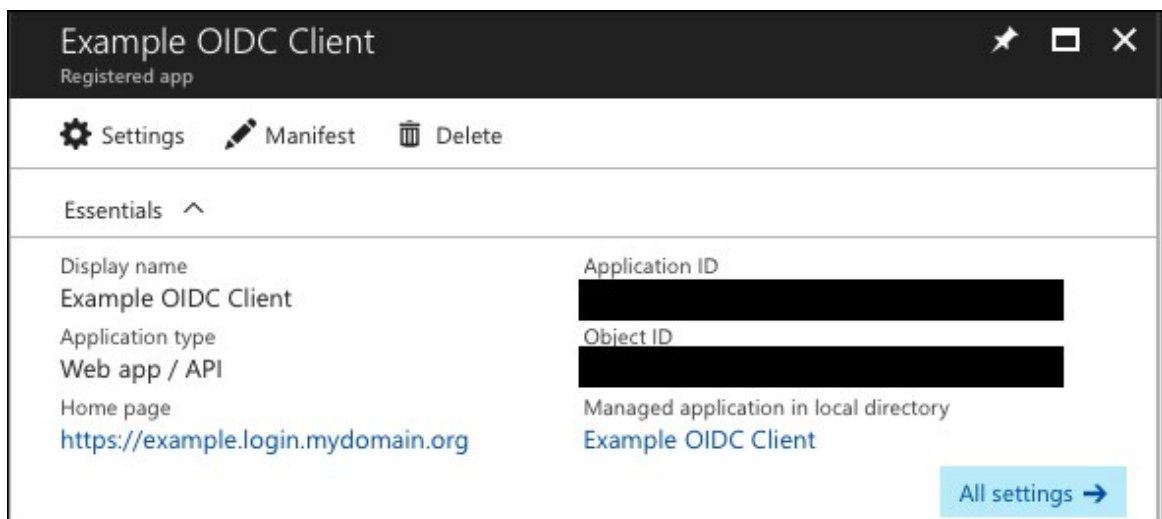
- Use the search bar to find your application registration, and click on its listing in the search results.



The interface shows a search bar with 'OIDC' entered and a dropdown menu set to 'All apps'. Below the search bar is a table of application registrations:

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
 Example OIDC Client	Web app / API	[Redacted]

- Record the **Application ID** displayed on the screen. This will be the **Relying Party OAuth Client ID**.

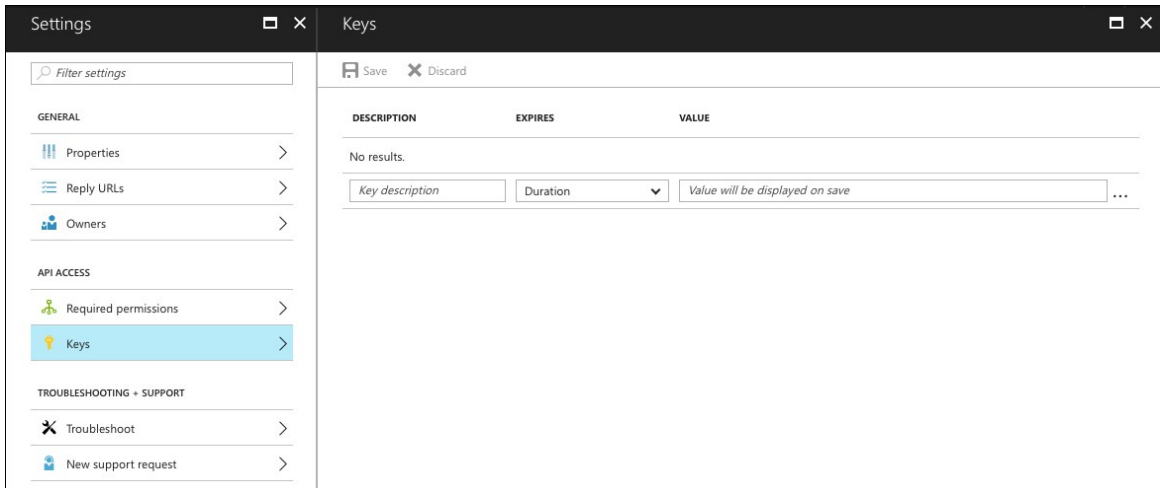


The 'Example OIDC Client' settings page shows the following details:

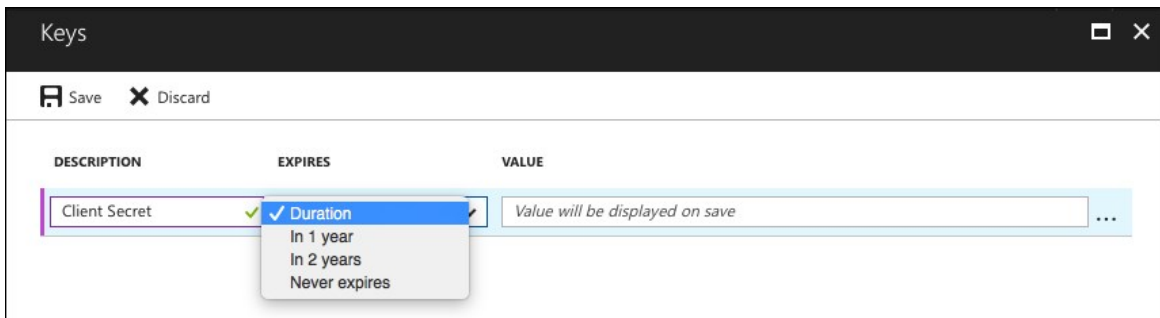
- Display name**: Example OIDC Client
- Application type**: Web app / API
- Home page**: https://example.login.mydomain.org
- Application ID**: [Redacted]
- Object ID**: [Redacted]
- Managed application in local directory**: Example OIDC Client

At the bottom right, there is a button labeled 'All settings →'.

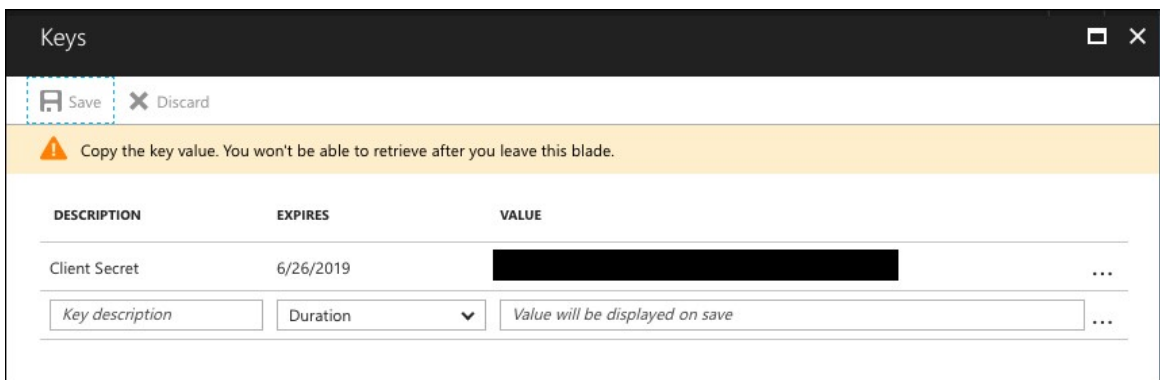
6. Open the **Keys** tab to generate your **Client Secret**.



7. Enter any name for the description of the key and select the appropriate duration for your security requirements.



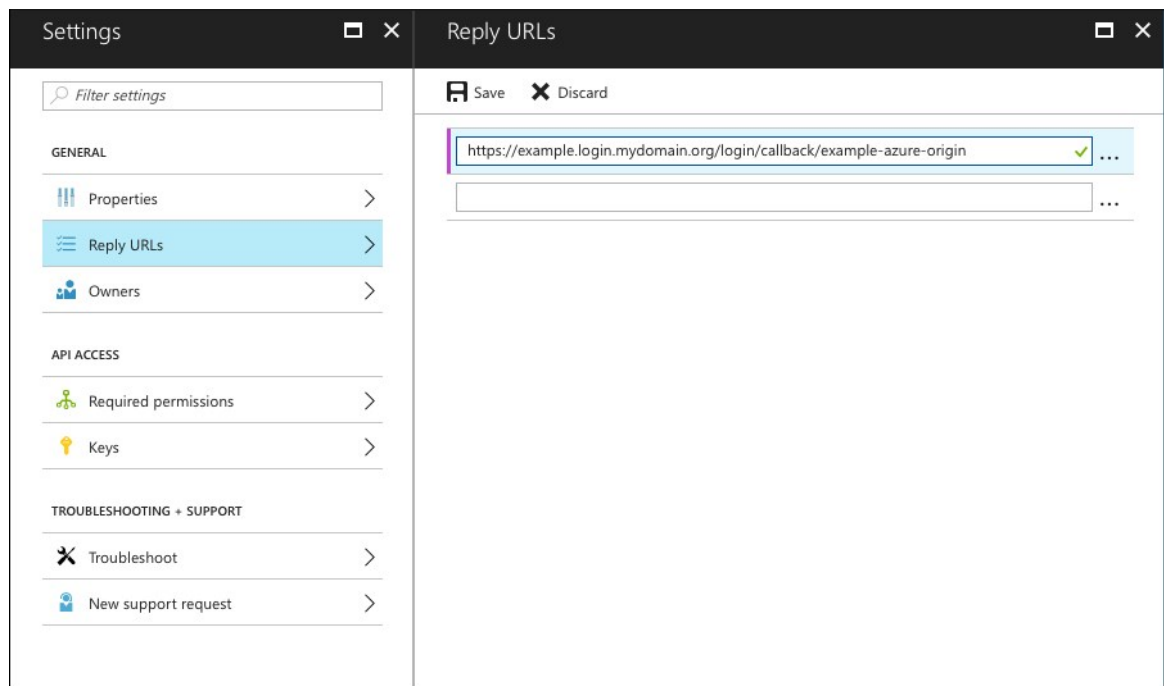
8. Click **Save** to generate your key value. This value is the **Relying Party OAuth Client Secret**. Record this value for future use.



9. Under **Reply URLs**, configure and save the URI of the form

https://AUTH_DOMAIN/login/callback/ORIGIN_KEY where:

- ✦ **AUTH_DOMAIN** is the Auth Domain setting you entered when you created the service plan that you are integrating with Azure AD.
- ✦ **ORIGIN_KEY** is based on the **Identity Provider Name** you set in the SSO dashboard in **Set Up OIDC Identity Provider in SSO** as shown below. Do not use spaces or uppercase letters in this value. You might need to change this later.



10. Identify your **Azure Tenant Name**. One location you can use to help you identify this is the **App ID URI** which uses the form `https://TENANT-NAME/APPLICATION-ID`.

For example, in the App ID URI `https://tenant.onmicrosoft.com/cj8472j2-d3d2-44b1-a2zf-ro5cd03f9584`, the Azure Tenant Name is `tenant.onmicrosoft.com`.

The screenshot displays two side-by-side windows from the VMware Tanzu Application Service interface. The left window, titled 'Settings', has a sidebar with a 'Filter settings' search bar and a list of categories: GENERAL, API ACCESS, and TROUBLESHOOTING + SUPPORT. Under GENERAL, 'Properties' is selected. The right window, titled 'Properties', contains configuration fields for an 'Example OIDC Client'. The 'App ID URI' field is highlighted with a red rectangle, showing a URL with redacted tenant and application identifiers. Other visible fields include Name, Object ID, Application ID, Logo (displaying 'EO'), Home page URL, Logout URL, Application type (set to 'Web app / API'), and a Multi-tenanted toggle set to 'No'.

11. Construct the URL for the OpenID Connect metadata endpoint by replacing `TENANT-NAME` with your Azure Tenant Name in the following string:

`https://login.microsoftonline.com/TENANT-NAME/.well-known/openid-configuration.`

Example: `https://login.microsoftonline.com/tenant.onmicrosoft.com/.well-known/openid-configuration`

Record these values for the [next step](#), configuring your OpenID Connect identity provider in SSO.

```
{
  "authorization_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/authorize",
  "token_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/token",
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "private_key_jwt",
    "jwks_uri"
  ],
  "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "http_logout_supported": true,
  "frontchannel_logout_supported": true,
  "end_session_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/logout",
  "response_types_supported": [
    "code",
    "id_token",
    "code id_token",
    "token id_token",
    "token"
  ],
  "scopes_supported": [
    "openid"
  ],
  "issuer": "https://sts.windows.net/TENANT-NAME/",
  "claims_supported": [
    "sub",
    "iss",
    "cloud_instance_name",
    "cloud_graph_host_name",
    "aud",
    "exp",
    "iat",
    "auth_time",
    "acr",
    "amr",
    "nonce",
    "email",
    "given_name",
    "family_name",
    "nickname"
  ],
  "microsoft_multi_refresh_token": true,
  "check_session_iframe": "https://login.microsoftonline.com/TENANT-NAME/oauth2/checksession",
  "userinfo_endpoint": "https://login.microsoftonline.com/TENANT-NAME/openid/userinfo",
  "tenant_region_scope": "NA",
  "cloud_instance_name": "microsoftonline.com",
  "cloud_graph_host_name": "graph.windows.net"
}

{
  "authorization_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/authorize",
  "token_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/token",
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "private_key_jwt",
    "jwks_uri"
  ],
  "jwks_uri": "https://login.microsoftonline.com/common/discovery/keys",
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "http_logout_supported": true,
  "frontchannel_logout_supported": true,
  "end_session_endpoint": "https://login.microsoftonline.com/TENANT-NAME/oauth2/logout",
  "response_types_supported": [
    "code",
    "id_token",
    "code id_token",
    "token id_token",
    "token"
  ],
  "scopes_supported": [
    "openid"
  ],
  "issuer": "https://sts.windows.net/TENANT-NAME/",
  "claims_supported": [
    "sub",
    "iss",
    "cloud_instance_name",
    "cloud_graph_host_name",
    "aud",
    "exp",
    "iat",
    "auth_time",
    "acr",
    "amr",
    "nonce",
    "email",
    "given_name",
    "family_name",
    "nickname"
  ],
  "microsoft_multi_refresh_token": true,
  "check_session_iframe": "https://login.microsoftonline.com/TENANT-NAME/oauth2/checksession",
  "userinfo_endpoint": "https://login.microsoftonline.com/TENANT-NAME/openid/userinfo",
  "tenant_region_scope": "NA",
  "cloud_instance_name": "microsoftonline.com",
  "cloud_graph_host_name": "graph.windows.net"
}
```

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Configuring an OIDC Service Provider in SSO



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how to add an OpenID Connect (OIDC) external identity provider to your Pivotal Single Sign-On (SSO) service plan, using Azure Active Directory (Azure AD) as an example.

Follow the steps below to set up an OIDC provider for the SSO service.

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 Application Service 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 Azure Origin` becomes `example-azure-origin`. If you did not enter this for your OAuth Client's authorized redirect URIs, [go back](#) and edit the value in Azure.
5. Enter a **Description**. Space developers see this description when they select an identity provider for their app.
6. Under **Identity Provider type**, select **OpenID Connect**.

New Identity Provider Cancel Create Identity Provider

Identity Provider Name*

Example Azure Origin

This name will show as a link on the login page

Identity Provider Description

Allows Azure User to authenticate.

Identity Provider type*

OpenID Connect

7. Clear the **Enable Discovery** checkbox and enter the following information from the **OpenID Connect metadata endpoint** you constructed at the end of the [previous section](#).
 - ✦ For **Authorization Endpoint URL**, enter in the `authorization_endpoint` value from the metadata endpoint.
 - ✦ For **Token Endpoint URL**, enter the `token_endpoint` value from the metadata endpoint.
 - ✦ For **Token Key**, enter the `jwks_uri` value from the metadata endpoint.
 - ✦ For **Issuer**, enter the `issuer` value from the metadata endpoint.
 - ✦ For **User Info Endpoint URL**, enter the `userinfo_endpoint` value from the metadata endpoint.
 - ✦ For **Response Type**, select `id_token`.
 - ✦ For **Relying Party OAuth Client ID**, enter the **Application ID** value recorded from the previous section.
 - ✦ For **Relying Party OAuth Client Secret**, enter the **Client Secret** value recorded in the previous section.

OpenID Connect Settings

☐ Skip SSL Validation

☐ Enable Discovery

Authorization Endpoint URL*

Token Endpoint URL*

Token Key*

Issuer

User Info Endpoint URL

Response Type

Relying Party OAuth Client ID*

Relying Party OAuth Client Secret

8. Select `openid` as a scope. You can select additional scopes.

Scopes*

☒ `openid`

9. Under **Advanced Settings** > **User Attributes**, map `user_name` to `unique_name`. This enables SSO to identify the authenticated user.
10. (Optional) Configure additional attribute mappings.
11. Click **Create Identity Provider** to save your settings.
12. (Optional) [Enable identity provider discovery](#) for the service plan.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing Your Single Sign-On Connection

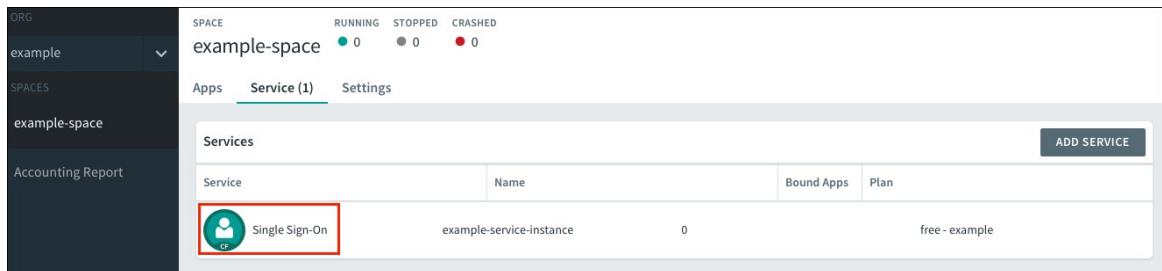


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 Azure Active Directory (Azure AD).

Follow the steps below to test your Single Sign-On connection.

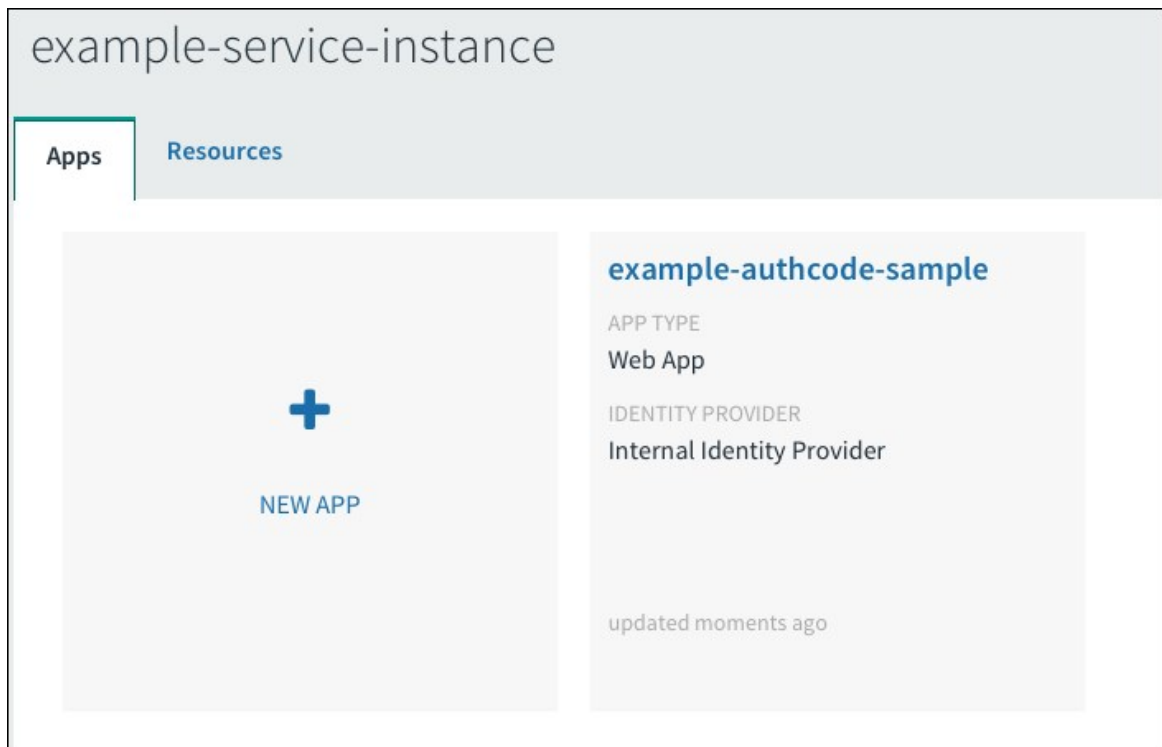
1. Log in to Apps Manager at <https://apps.YOUR-SYSTEM-DOMAIN> and navigate to the org and space where your app is located.
2. Under **Services**, locate the service instance of the SSO plan bound to your app.



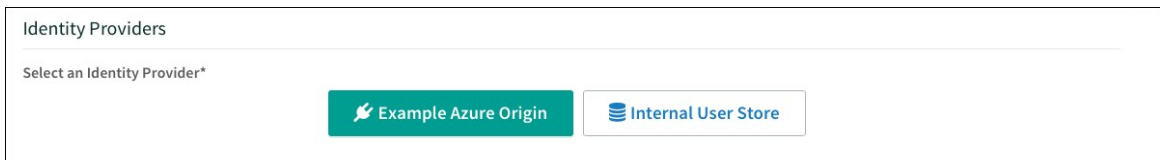
3. Select the service instance and click **Manage**.



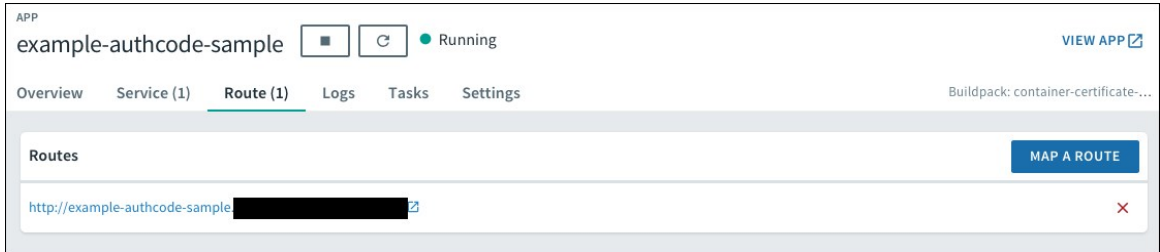
4. Under the **Apps** tab, select your app.



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



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



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



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

Example OIDC Client

Work or school, or personal Microsoft account

☐ Keep me signed in

Sign in

Back

[Can't access your account?](#)

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

Example OIDC Client

App publisher website: [REDACTED]onmicrosoft.com

Example OIDC Client needs permission to:

- Sign you in and read your profile ?

You're signed in as: [REDACTED]

[Show details](#)

Accept

Cancel

10. If you are now logged into your app, your Azure AD OIDC to SSO connection works.

Authcode sample

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

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Troubleshooting

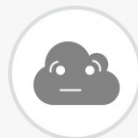


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Bad Request

Symptom:



There was an error when authenticating against the external identity provider: 400 Bad Request

Explanations:

- This is a generic error. Review UAA logs for detailed information.
- This error can occur when the application type is created as **Native**. Ensure you created your client in Azure AD as **Web App/API**.
- This error can occur when a response type other than `id_token` is used. Ensure you configure the response type to use `id_token`.

Cannot determine username from credentials supplied

Symptom:



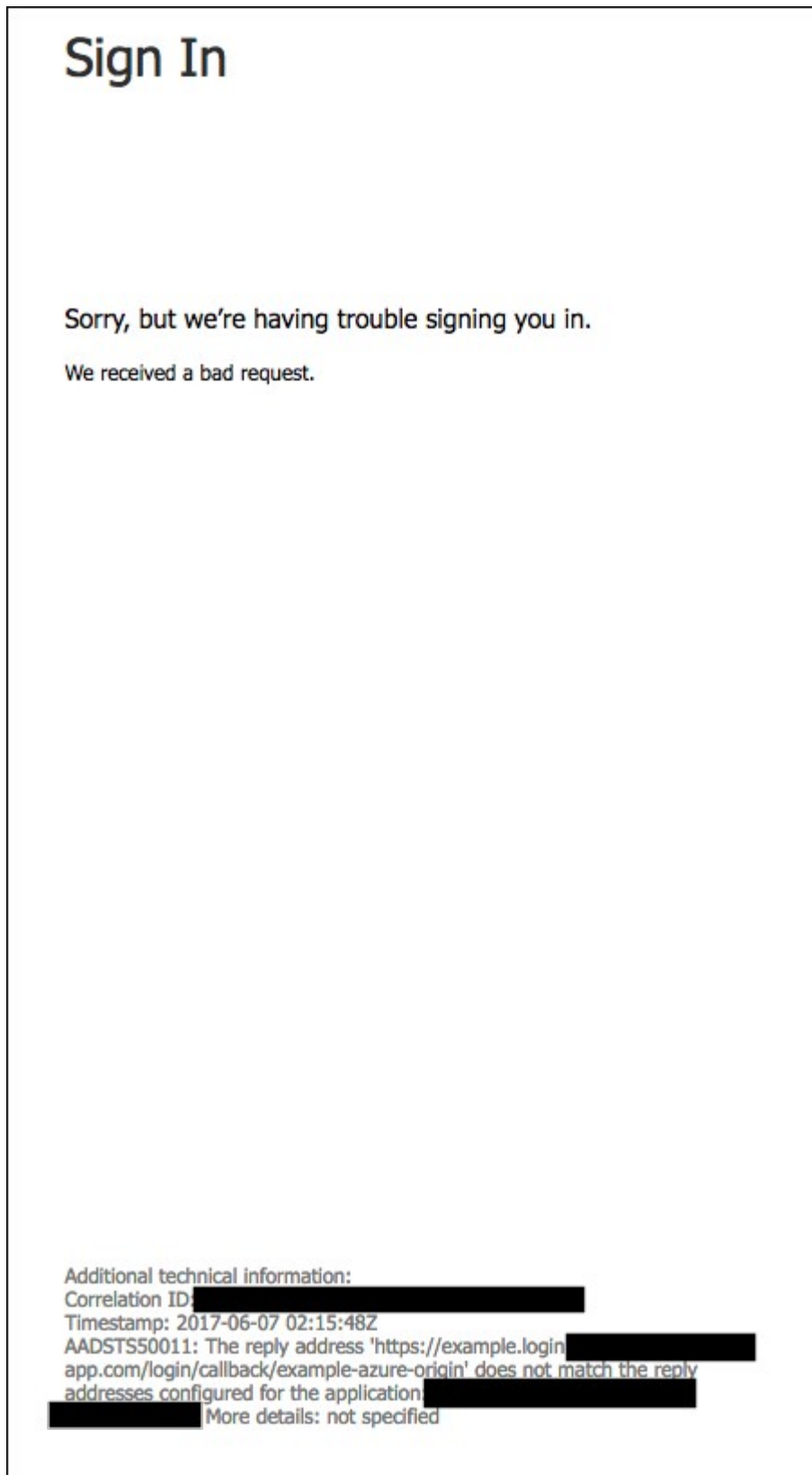
There was an error when authenticating against the external identity provider: Cannot determine username from credentials supplied

Explanation:

- No value is mapped to the username used by PCF. Under the identity provider attributes, map the `unique_name` attribute to `username`

Azure Error for Reply Address

Symptom:



Explanation:

- The reply URL is misconfigured. Ensure you entered your callback URL correctly as a reply URL in Azure AD.

Login Page Cannot Be Found (404 Error)

Symptom:



This **login.windows.net** page can't be found

Explanation:

- The Authorization Endpoint URL may be incorrectly entered or not available. Ensure you correctly entered the authorization endpoint, and that the authorization endpoint is available to the end user.

Error authenticating against external identity provider: 404 Not Found

Symptom:



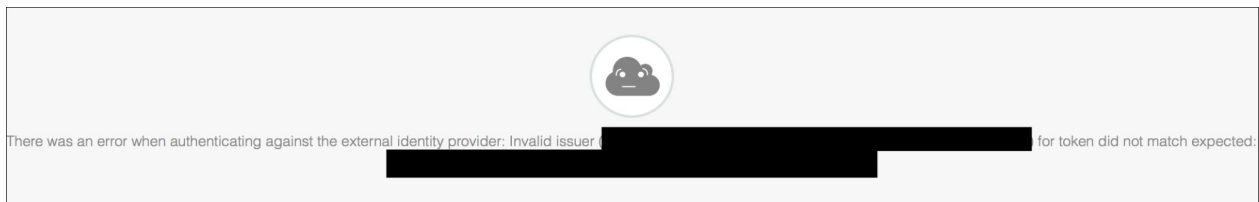
There was an error when authenticating against the external identity provider: 404 Not Found

Explanation:

- The Token Key URL may be incorrectly entered or not available. Ensure that you entered the token key setting correctly, and that the Token Key URL is available.

Error authenticating against external identity provider: Invalid issuer for token did not match expected

Symptom:

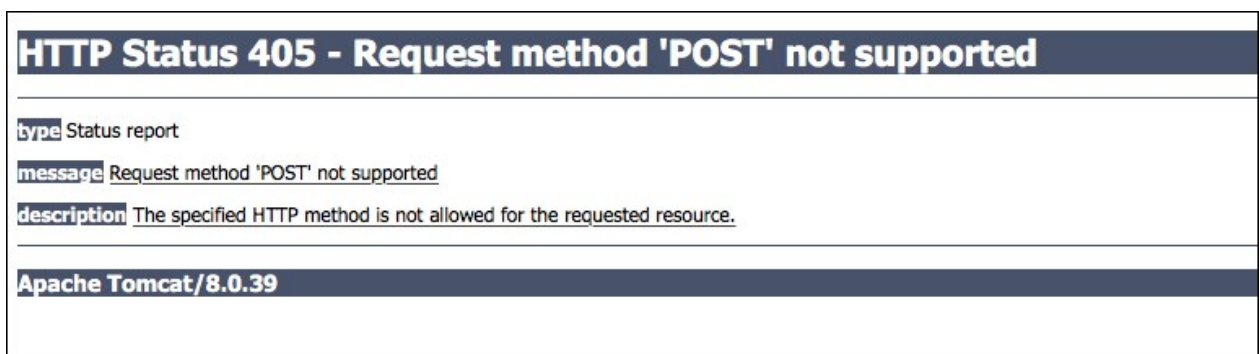


Explanation:

- The Token Key URL may be incorrectly entered. Ensure that you entered the issuer setting correctly.

Request Method 'POST' not supported (405 Error)

Symptom:

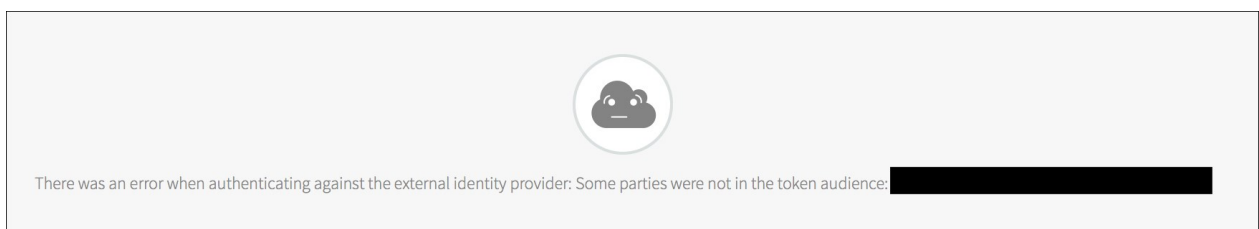


Explanation:

- This error can occur if you configure a response type that Azure AD does not support or has not been enabled for the application, such as `token` or `code id_token token`. Ensure that you configure the response type to `id_token`.

Error authenticating against external identity provider: Some parties were not in the token audience

Symptom:



Explanation:

- The Relying Party Client ID may be incorrectly entered. Ensure you have correctly entered the relying party client ID setting.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

CA Single Sign-On Integration Guide

CA Single Sign-On Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

CA Single Sign-On

- CA Single Sign-On, v12.52 or later.
- A Signed Certificate by a Certificate Authority.



Note: To configure SAML, you must have the Pivotal 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring CA Single Sign-On as an Identity Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:
 1. Select **Local** for **Entity Location**.
 2. Select **SAML2 IDP** for **New Entity Type**.
 3. Click the **Next** button.
5. In the **Entities** section, perform the following steps:
 1. Enter an **Entity ID**.
 2. Enter an **Entity Name**.
 3. Enter a **Description**.
 4. Enter the fully-qualified domain name for your CA Single Sign-On as the **Base URL**.
 5. Select or import a **Signing Private Key Alias**.
 6. Select a **Name ID format**.
 7. Click the **Next** button.
6. Confirm the Entity Details and click the **Finish** button.

The screenshot shows the 'Entities' configuration page in the CA Single Sign-On interface. The page is titled 'Entities' and has a breadcrumb 'View Federation Entities > View Entity'. A 'Return to View Federation Entities' link is in the top right. The 'Entity Type' section shows 'Entity Location: Local' and 'Entity Type: SAML2 IDP'. The 'Entity Details' section contains the following fields and values:

- Entity ID: smidp
- Entity Name: smidp
- Description:
- Base URL: https://sc5.casecorecenter.com
- Default SLO Confirm URL: https://sc5.casecorecenter.com
- SOAP Artifact Resolution URL: https://sc5.casecorecenter.com/affwebservices/public/saml2ars
- SSO Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2sso
- SLO Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2slo
- SLO SOAP Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2slosoap
- User Consent Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2userconsent
- Attribute Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2attrsvc
- SOAP Manage NameID Service URL: https://sc5.casecorecenter.com/affwebservices/public/saml2nidsoap

The 'Default Signature and Encryption Options' section shows 'Signing Private Key Alias: signingcert' and 'Signed Authentication Requests Required: No'. The 'Supported Name ID Formats and Attributes' section has a table with 'Supported Name ID Formats' and 'Supported Assertion Attributes'.

Supported Name ID Formats		Supported Assertion Attributes	
Selected Formats		Assertion Attribute	Supported Format
Email Address			
Unspecified			

7. Click the **Federation** tab.
8. Click on the **Entities** link.
9. Click the **Import Metadata** button and perform the following steps:
 1. Click **Browse** and select the downloaded metadata for **Metadata file**.

2. Select **Remote Entity** for **Import As**.
 3. Select **Create New** for **Operation**.
 4. Click the **Next** button.
10. In the **Select Entity Defined in Metadata File** section, perform the following steps:
 1. Enter an **Entity Name**.
 2. Click the **Next** button.
 11. In the **Select Key Entries to Import** section, perform the following steps:
 1. Enter an **Alias**.
 2. Click the **Next** button.
 12. Confirm the Entity Details and click the **Finish** button.

Entities

[View Federation Entities](#) + [View Entity](#) [Return to View Federation Entities](#)

Entity Type

Entity Location: Remote
Entity Type: SAML2 SP

Entity Details

Entity ID: http://sso.login.coral.springapps.io
Entity Name: pcf-coral
Description: pcf-coral imported via metadata

Remote Assertion Consumer Service URLs

Index	Binding	URL	Default
0	HTTP-POST	https://sso.login.coral.springapps.io/saml/SSO/alias/sso.login.coral.springapps.io	Yes
1	HTTP-Artifact	https://sso.login.coral.springapps.io/saml/SSO/alias/sso.login.coral.springapps.io	No

Remote SLO Service URLs

Binding	Location URL	Response Location URL
HTTP-POST	https://sso.login.coral.springapps.io/saml/SingleLogout/alias/sso.login.coral.springapps.io	
HTTP-Redirect	https://sso.login.coral.springapps.io/saml/SingleLogout/alias/sso.login.coral.springapps.io	

Manage Name ID Service URLs

Binding	Location URL	Response Location URL

Signature and Encryption Options

Verification Certificate Alias: pcf-coral
Encryption Certificate Alias: pcf-coral
Sign Authentication Requests: Yes

Name ID Formats

Supported Name ID Formats	Selected Formats

(Email Address)

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:
 1. Enter a **Partnership Name**.
 2. Enter a **Description**.
 3. Select a previously created local entity for **Local IDP**.
 4. Select a previously created remote entity for **Remote SP**.
 5. Enter a **Skew Time**.
 6. Add any **User Directories**.
 7. 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:
1. Select a **Name ID Format**.
 2. Select **User Attribute** as the **Name ID Type**.
 3. Enter `mail` as the **Value**.
 4. (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**.

5. Click the **Next** button.

Assertion Attributes	Retrieval Method	Format	Type	Value	DN Spec	Encrypt
roles	SSO	URI	User Attribute	FMATTR:SM_USERGROUPS	No	No
mail	SSO	URI	User Attribute	mail	No	No

18. In the **SSO and SLO** section, perform the following steps:
 1. Enter the **Authentication URL**.
 2. Select **HTTP-Post** for **SSO Binding**.
 3. Select **Both IDP and SP initiated** for **Transactions Allowed**.
 4. Click the **Next** button.

19. In the **Signature and Encryption** section, perform the following steps:
 1. Select your key alias for **Signing Private Key Alias**.
 2. Select your certificate alias for **Verification Certificate Alias**.
 3. Click the **Next** button.

20. Confirm the Partnership Details and click the **Finish** button.
21. Click the **Action** button and click **Activate**.

Federation Partnership List							Create Partnership
Actions	Name	Local Type	Local Entity ID	Remote Type	Remote Entity ID	Status	FIPS Status
Action	pcf-coral-ssn	SAML2 IDP	smidp	SAML2 SP	http://sso.login.coral.springapps.io	Defined	✗
Action		IDP	smidp	SAML2 SP	https://myclouddemo-dev-ed.my.salesforce.com	Active	✗
Action		IDP	smidp	SAML2 SP	ssotest.login.run.pivotal.io	Active	✗

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

Create a pull request or raise an issue on the source for this page in GitHub

Configuring a Single Sign-On Service Provider

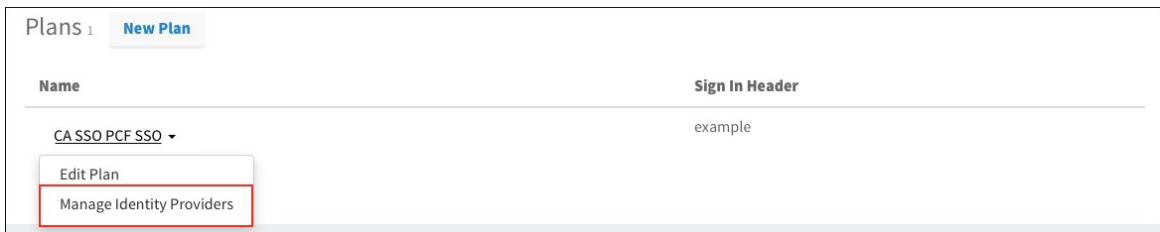


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Setting up SAML

- Log in to the SSO dashboard at <https://p-identity.YOUR-SYSTEM-DOMAIN> as a Plan Administrator.
- Select your plan and click **Manage Identity Providers** on the drop-down menu.



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

New Identity Provider

Identity Provider Name*

This name will show as a link on the login page

Identity Provider Description

Allows to authenticate.

Identity Provider Type*

SAML

Identity Provider Metadata

Identity Provider Metadata URL*

Fetch Metadata

▶ SAML File Metadata (optional)

Advanced SAML Settings

▶ Attribute mappings (optional)

Cancel **Create Identity Provider**

4. To create a new identity provider, perform the following steps:
 1. Enter an identity provider name in **Identity Provider Name**.
 2. (Optional) Enter a description in **Identity Provider Description**.
 3. Click **SAML File Metadata (optional)** followed by clicking the **Upload Identity Provider Metadata** button to upload your metadata XML.
 4. (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:
 1. Enter a **Group Name**.
 2. 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has


reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

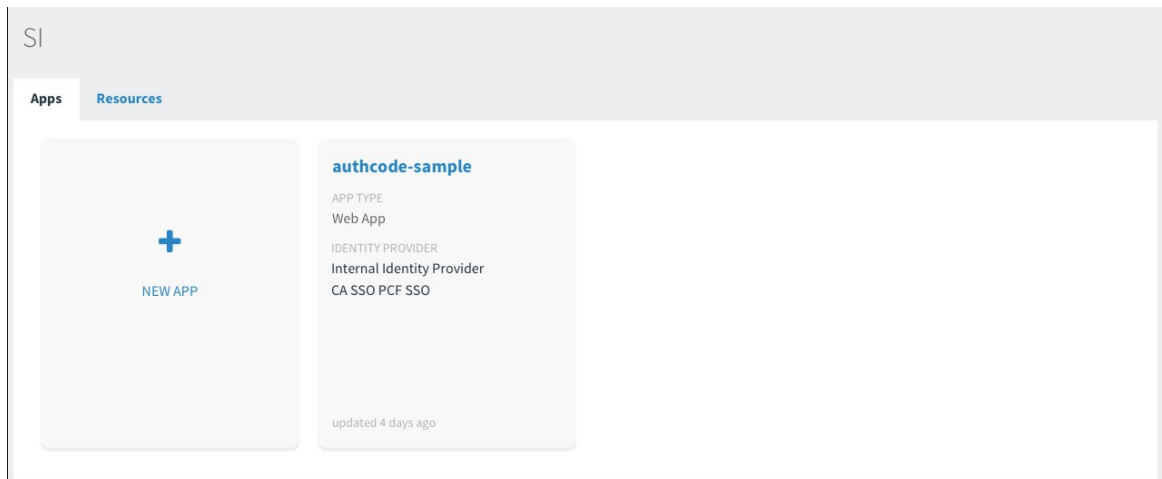
1. Log in to Apps Manager at <https://apps.YOUR-SYSTEM-DOMAIN> and navigate to the organization and space where your application is located.
2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Select the service instance and click **Manage**.

The screenshot shows the 'Services' tab in the Apps Manager interface. It features a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. The 'Pivotal Single Sign-On' service instance is highlighted with a red box. The table also includes an 'Add Service' button.

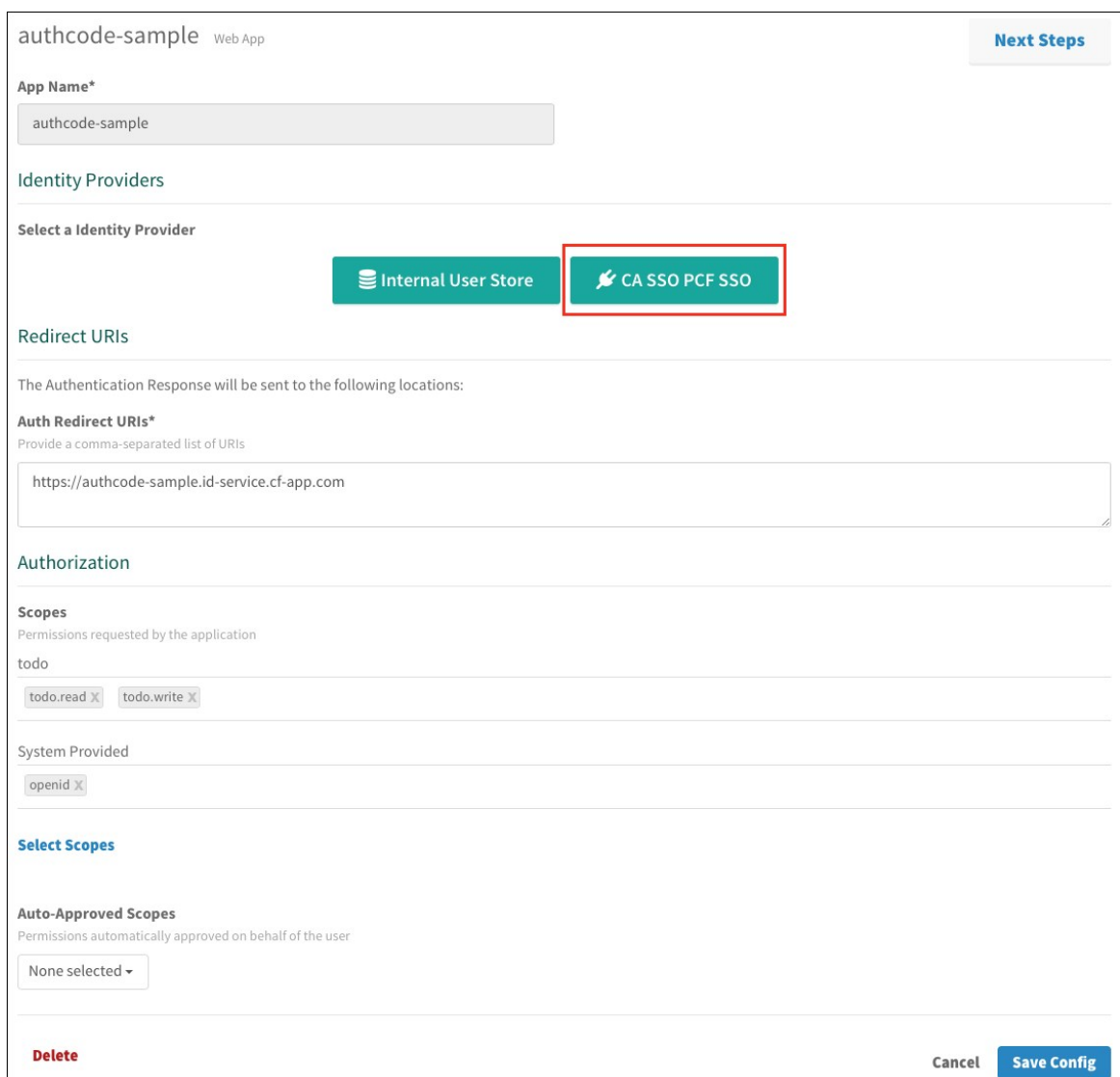
SERVICE	NAME	BOUND APPS	PLAN
	Pivotal Single Sign-On	1	free - (MONTHLY)

The screenshot shows the details page for the 'Pivotal Single Sign-On' service instance. It includes tabs for 'App Binding (1)', 'Plan', and 'Settings'. The 'Manage' button is highlighted with a red box. The page also displays the service plan 'CA SSO PCF SSO' and a 'Bound Apps' section with the application 'authcode-sample'.

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



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



authcode-sample Web App Next Steps

App Name*

authcode-sample

Identity Providers

Select a Identity Provider

Internal User Store **CA SSO PCF SSO**

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected ▾

Delete Cancel Save Config

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

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap... >

6. Click the link.



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

Please Login

Username:

Password:

Login

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

Application Authorization

authcode-sample

<https://authcode-sample>

authcode-sample has requested permission to access your account. If you do not recognize this application or its URL, you should click deny. The application will not see your password.

☒ openid

☒ Read TODO list

You can change your approval of permissions or revoke access for this application at any time from account settings. By approving access, you agree to authcode-sample's terms of service and privacy policy.

DENY
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" ],
  "token_type" : "bearer",
  "expires_in" : 3600
}
```

```

"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.

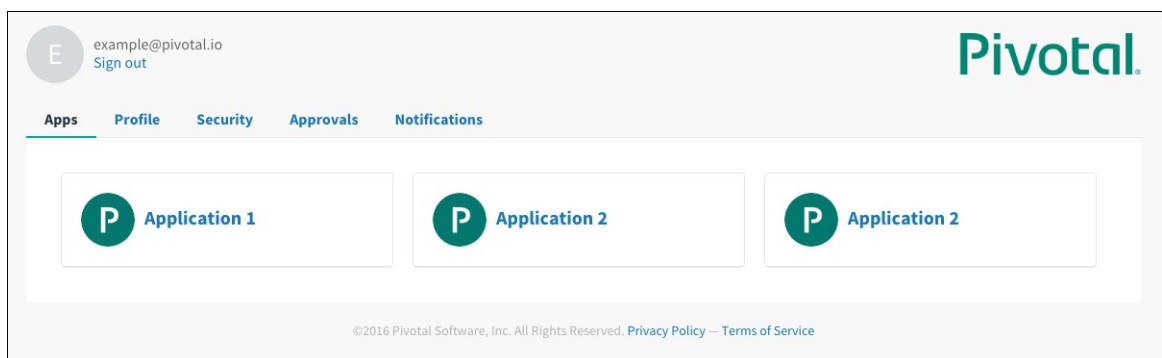


Please Login

Username:

Password:

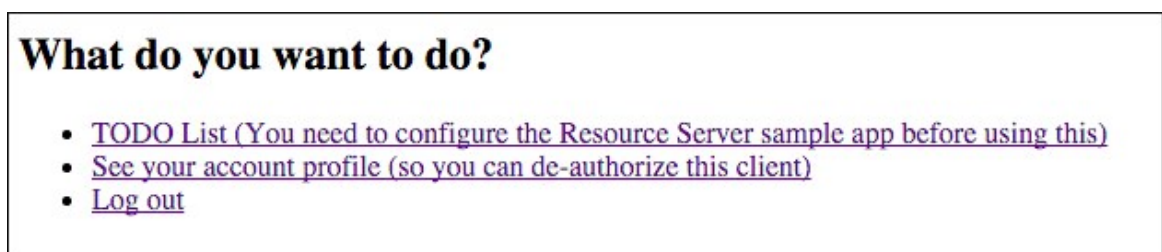
2. Navigate to your application and click it.
3. You are redirected to the page that lists applications you have access to.



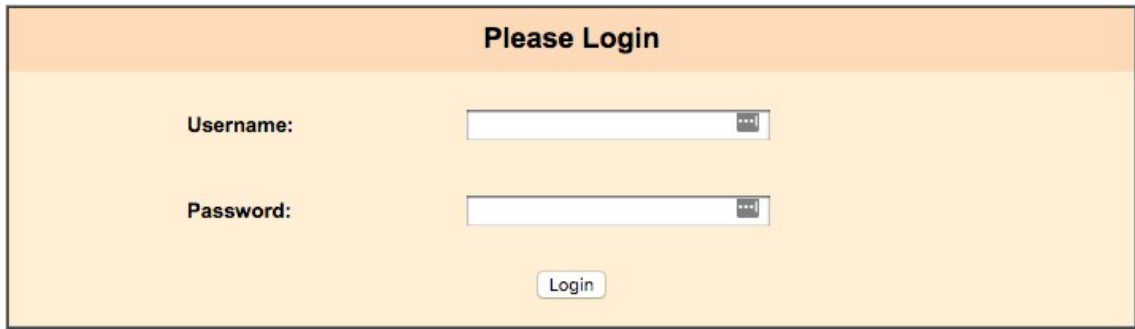
Test Your Single Sign-Off

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.



Please Login

Username:

Password:

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Troubleshooting



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:

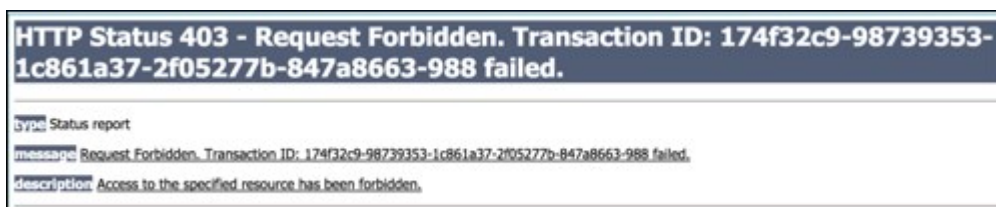
The following error occurred: 403 - Request Forbidden. Transaction ID: d5dd24e-950bf795-1a3cf7c7-0bcb12dc-82689d7c-bc failed.

Explanations:

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

Service Provider Entity ID Misconfigured

Symptom:

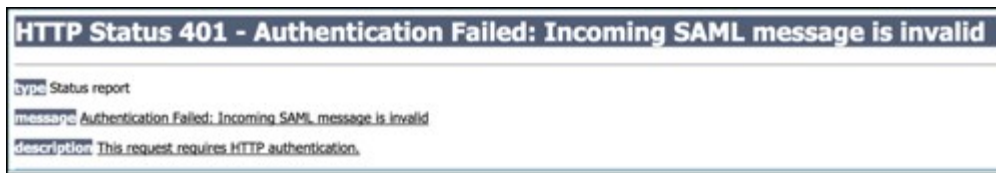


Explanation:

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

Incoming SAML message is invalid

Symptom:

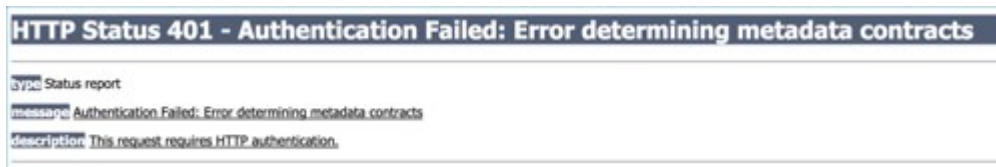


Explanation:

- 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:

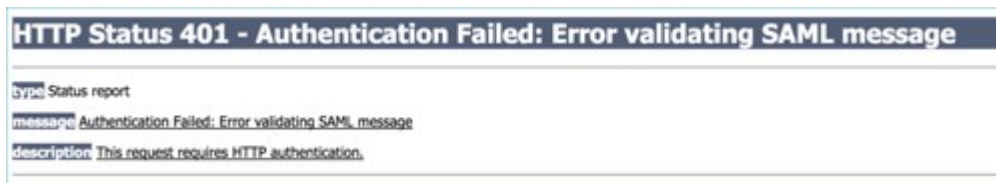


Explanation:

- 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:

The following error occurred: 500 - Internal Error occurred while trying to process the request. Transaction ID: 276f2b31-154b7e4b-383ebaf0-7ee1a10f-e1c3d4

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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Google Cloud Platform OIDC Integration Guide

Google Cloud Platform OIDC Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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 GCP as a single sign-on identity provider for PCF apps, you need:

Pivotal

- PCF, v1.11.0 or later.
- Single Sign-On, 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](#).

GCP

- 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring GCP as an OIDC Identity Provider

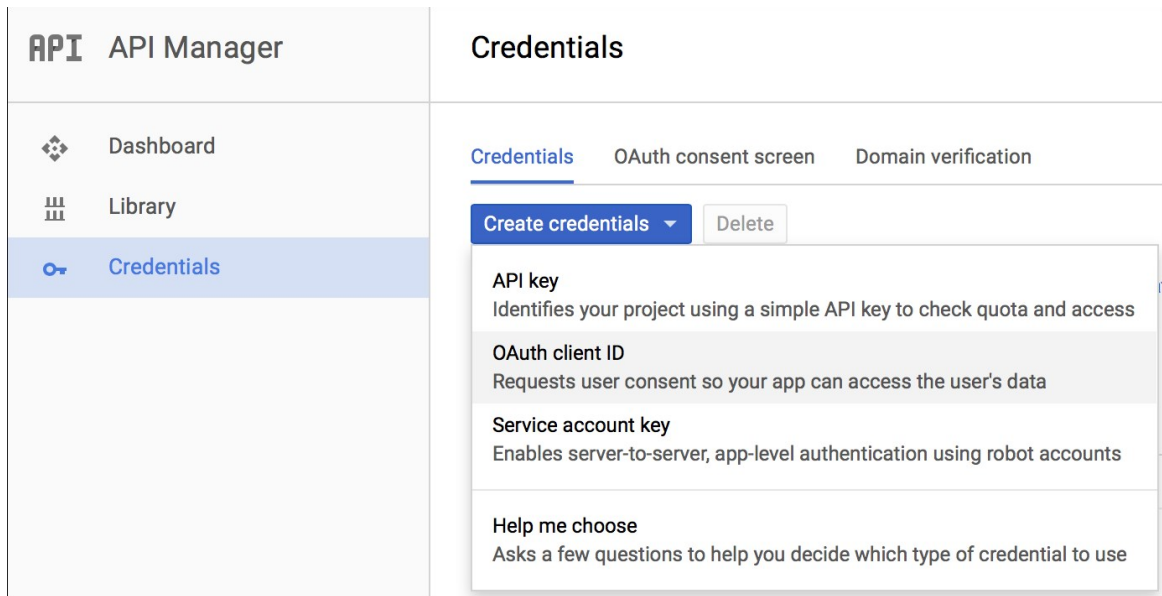


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

Application type

☒ Web application
☐ Android [Learn more](#)
☐ Chrome App [Learn more](#)
☐ iOS [Learn more](#)
☐ PlayStation 4
☐ Other

Name

OAuth Client Example

Restrictions
Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (http://*.example.com) or a path (http://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

http://www.example.com

Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://example.login.mydomain.org/login/callback/example-google-origin ×


http://www.example.com/oauth2callback

Create **Cancel**


- 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](#).

OAuth client

Here is your client ID

[Redacted client ID] 

Here is your client secret

[Redacted client secret] 

OK

Set Up OIDC Identity Provider in SSO

- 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 Application Service tile in Ops Manager under the **Credentials** tab.
- 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**.

New Identity Provider Cancel Create Identity Provider

Identity Provider Name*

Example Google Origin

This name will show as a link on the login page

Identity Provider Description

Allows Google User to authenticate.

Identity Provider type*

OpenID Connect

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.

OpenID Connect Settings

☐ Skip SSL Validation

☒ Enable Discovery

Discovery Endpoint URL*

https://accounts.google.com/.well-known/openid-configuration

Fetch Scopes

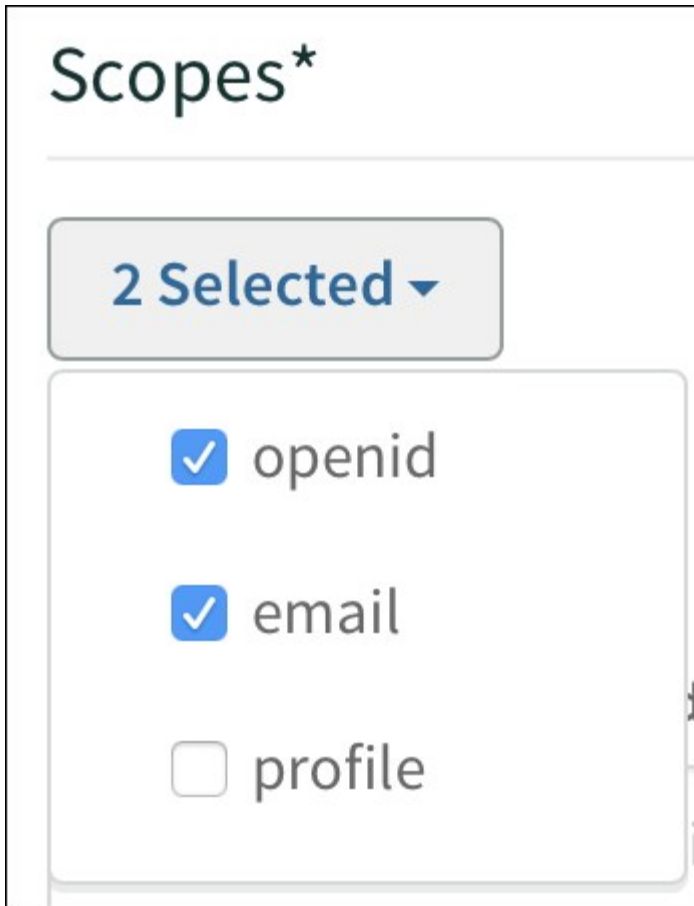
Relying Party OAuth Client ID*

your-client-id-here

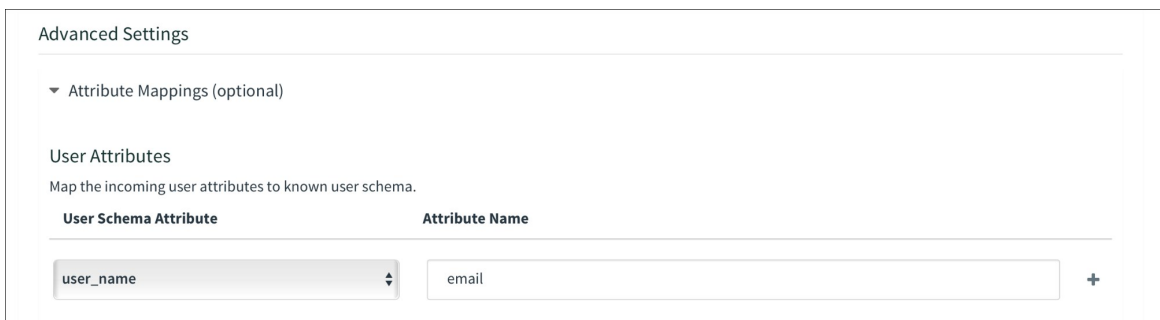
Relying Party OAuth Client Secret*

.....

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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing

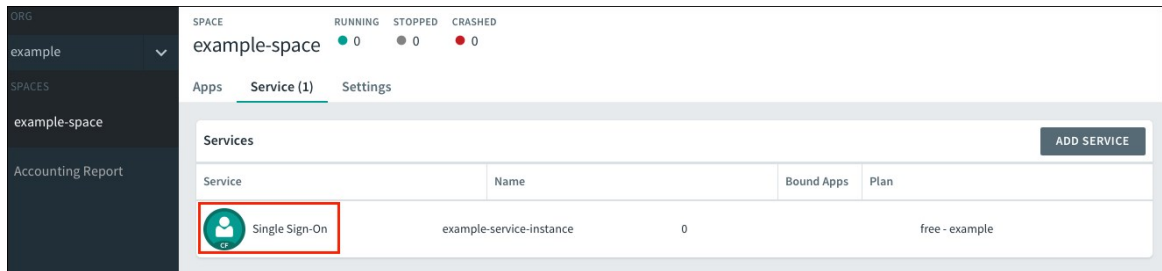


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

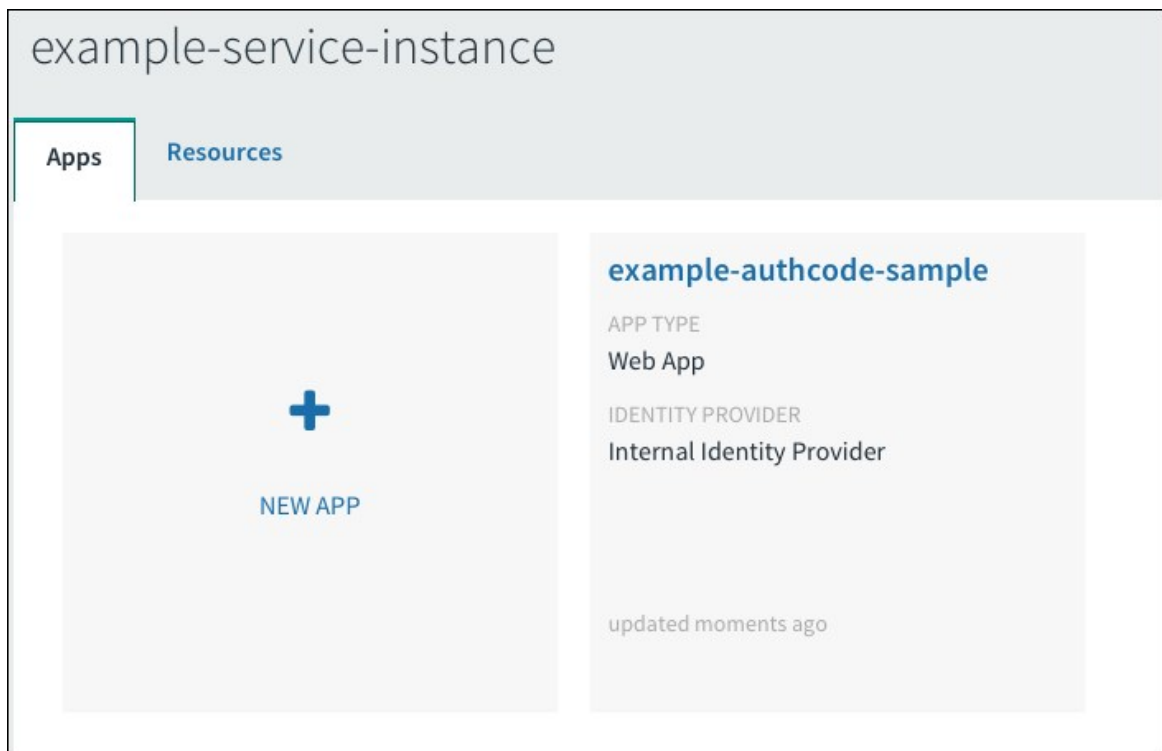
1. Log in to Apps Manager at <https://apps.YOUR-SYSTEM-DOMAIN> and navigate to the org and space where your app is located.
2. Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your app.



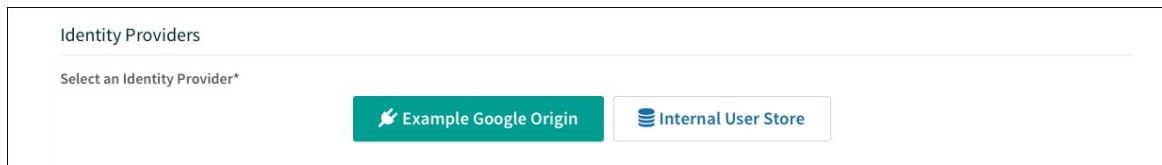
3. Select the service instance and click **Manage**.



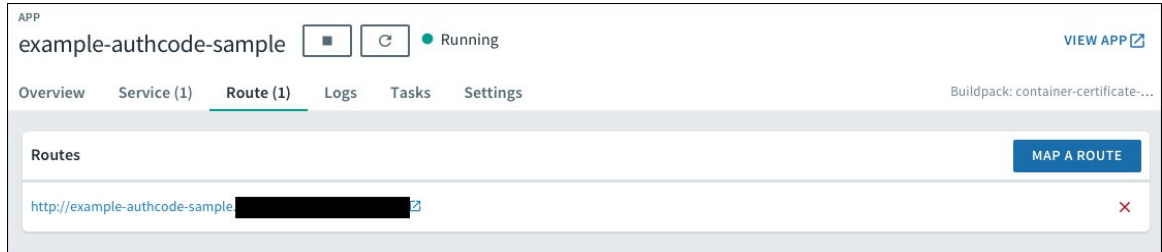
4. Under the **Apps** tab, select your app.



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



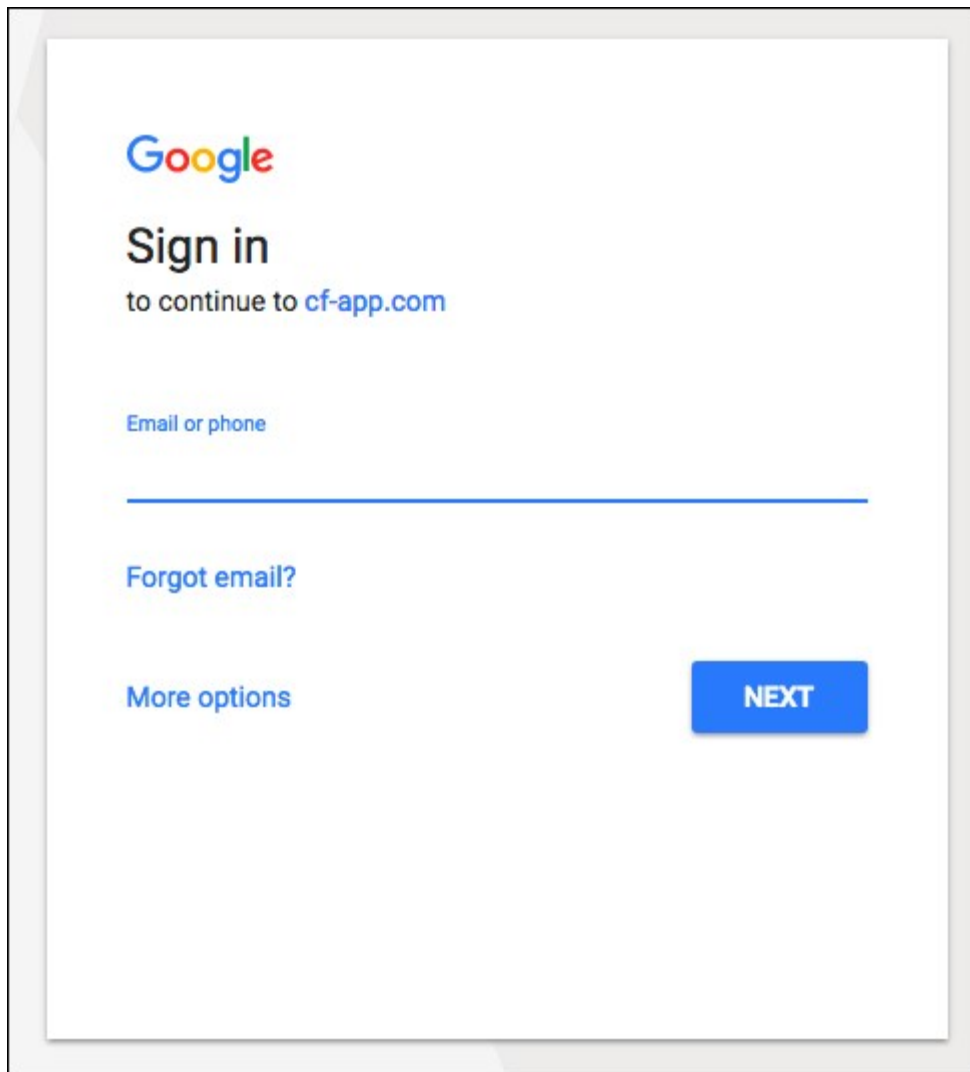
6. Return to Apps Manager and click the URL listed below your app to access your application.



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.

Authcode sample

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Troubleshooting



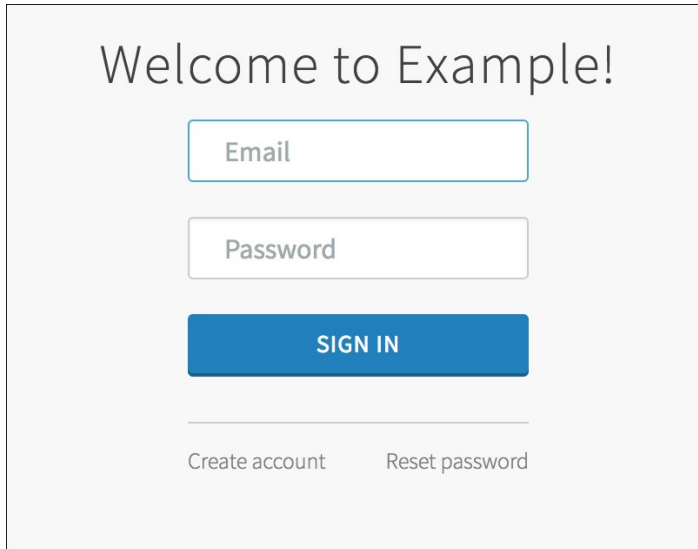
Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:



Welcome to Example!

Email

Password

SIGN IN

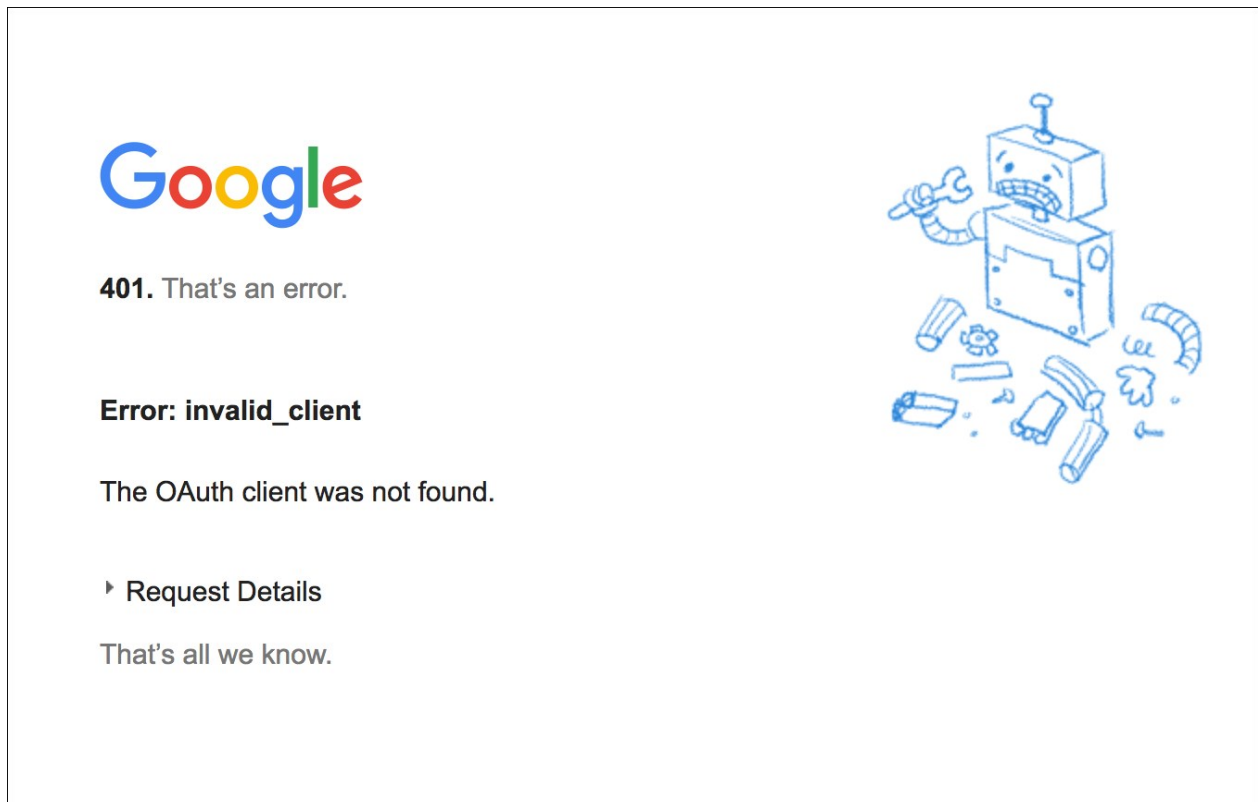
Create account Reset password

Explanation:

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

No OAuth Client Found

Symptom:

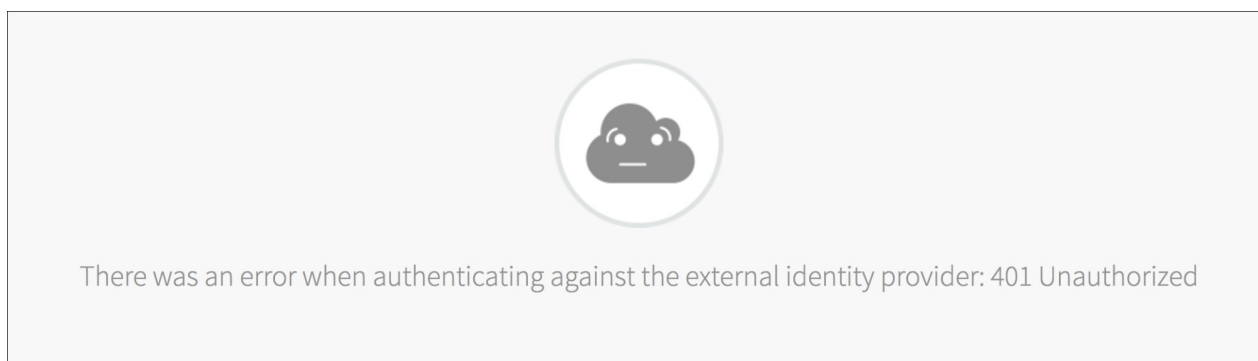


Explanation:

- Incorrect OAuth Client ID configured.

Unauthorized

Symptom:



Explanation:

- Incorrect OAuth client secret configured.

Redirect URI Mismatch

Symptom:



400. That's an error.

Error: redirect_uri_mismatch

The redirect URI in the request, [https://example.login\[REDACTED\]/login/callback/example-google-origin](https://example.login[REDACTED]/login/callback/example-google-origin), does not match the ones authorized for the OAuth client. Visit [https://console.developers.google.com/apis/credentials/oauth\[REDACTED\]](https://console.developers.google.com/apis/credentials/oauth[REDACTED]) to update the authorized redirect URIs.

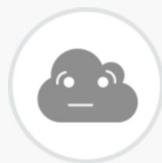


Explanation:

- Incorrect authorization redirect URI on OAuth Client.

Empty Username

Symptom:



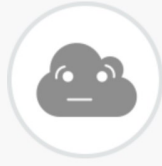
There was an error when authenticating against the external identity provider: Username cannot be empty

Explanation:

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

Unable to map claim to a username

Symptom:



There was an error when authenticating against the external identity provider: Username cannot be empty

Explanation:

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Okta Integration Guide

Okta Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

Okta

- Okta, v2016.07 or later.
- A user with Application Admin privileges.



Note: To configure SAML, you must have the Pivotal 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](#)

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Configuring Okta as an Identity Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

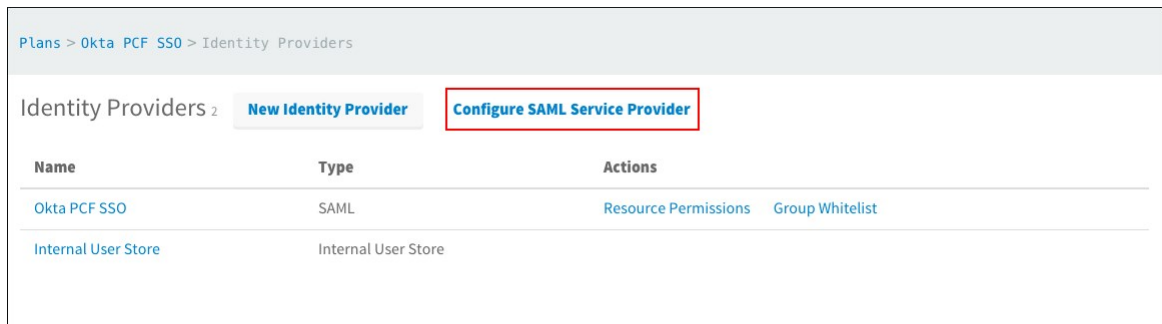
This topic describes how to set up Okta as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and Okta.

Set up SAML in PCF

1. Log into the SSO dashboard at <https://p-identity.YOUR-SYSTEM-DOMAIN> using your User Account and Authentication (UAA) administrator credentials. In your Pivotal Application Services tile in Ops Manager, the **Domain** settings shows your system domain, and the **Credentials** tab shows the **UAA Admin Credentials**.
2. Select your plan and click **Manage Identity Providers** on the drop-down menu.



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



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



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

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

The screenshot shows the Okta PCF SSO configuration interface. At the top, there's a header with the Okta logo, a gear icon, and the text 'Okta PCF SSO'. Below this is a navigation bar with tabs: General, Sign On (selected), Mobile, Import, People, and Groups. The main content area is titled 'Settings' and has an 'Edit' button. Under 'SIGN ON METHODS', there's a section for 'SAML 2.0' which is highlighted with a red box. Below this, there's a yellow warning box stating 'SAML 2.0 is not configured until you complete the setup instructions.' and a 'View Setup Instructions' button. At the bottom, there's a 'CREDENTIALS DETAILS' section with options for 'Application username format' (Okta username) and 'Password reveal' (Allow users to securely see their password (Recommended)). On the right side, there's an 'About' section explaining SAML 2.0 and an 'Application Username' section with instructions on choosing a format.

Okta PCF SSO

← Back to Applications

Active

General Sign On Mobile Import People Groups

Settings Edit

SIGN ON METHODS

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

SAML 2.0

Default Relay State

SAML 2.0 is not configured until you complete the setup instructions.

View Setup Instructions

Identity Provider metadata is available if this application supports dynamic configuration.

CREDENTIALS DETAILS

Application username format Okta username

Password reveal ☐ Allow users to securely see their password (Recommended)

About

SAML 2.0 streamlines the end user experience by not requiring the user to know their credentials. Users cannot edit their credentials when SAML 2.0 is configured for this application. Additional configuration in the 3rd party application may be required to complete the integration with Okta.

Application Username

Choose a format to use as the default username value when assigning the application to users.

If you select None you will be prompted to enter the username manually when assigning an application with password or profile push provisioning features.

4. Click the **General** tab.
5. Under **SAML Settings**, click the **Edit** button followed by the **Next** button.

Edit SAML Integration

1 General Settings 2 **Configure SAML** 3 Feedback

A SAML Settings

GENERAL

Single sign on URL

☒ Use this for Recipient URL and Destination URL

Audience URI (SP Entity ID)

Default RelayState

If no value is set, a blank RelayState is sent

Name ID format

Application username

[Show Advanced Settings](#)

ATTRIBUTE STATEMENTS (OPTIONAL) [LEARN MORE](#)

Name	Name format (optional)	Value
firstName	Unspecified	user.firstName
lastName	Unspecified	user.lastName
email	Unspecified	user.email

[Add Another](#)

GROUP ATTRIBUTE STATEMENTS (OPTIONAL)

Name	Name format (optional)	Filter
	Unspecified	Starts with

[Add Another](#)

What does this form do?
This form generates the XML needed for the app's SAML request.

Where do I find the info this form needs?
The app you're trying to integrate with should have its own documentation on using SAML. You'll need to find that doc, and it should outline what information you need to specify in this form.

Okta Certificate
Import the Okta certificate to your Identity Provider if required.

[Download Okta Certificate](#)

6. In the **SAML Settings** section:

1. 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`.
2. 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.
3. Select a **Name ID format**.
4. Select an **Application username**.

7. (Optional) To configure single logout:

1. Click **Show Advanced Settings**.
2. For **Enable Single Logout**, select **Allow application** to initiate single logout.
3. Enter the **SingleLogoutService Location URL** from your downloaded service provider metadata into **Single Logout URL**.
4. Enter your **Auth Domain URL** into **SP Issuer**.
5. 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](#).

Okta PCF SSO

Active

General Sign On Mobile Import People Groups

Settings Edit

SIGN ON METHODS

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

SAML 2.0

Default Relay State

SAML 2.0 is not configured until you complete the setup instructions.

View Setup Instructions

Identity Provider metadata is available if this application supports dynamic configuration.

CREDENTIALS DETAILS

Application username format Okta username

Password reveal ☐ Allow users to securely see their password (Recommended)

About

SAML 2.0 streamlines the end user experience by not requiring the user to know their credentials. Users cannot edit their credentials when SAML 2.0 is configured for this application. Additional configuration in the 3rd party application may be required to complete the integration with Okta.

Application Username

Choose a format to use as the default username value when assigning the application to users.

If you select None you will be prompted to enter the username manually when assigning an application with password or profile push provisioning features.

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Configuring a Single Sign-On Service Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

Name	Sign In Header
Okta PCF SSO ▾	example

Edit Plan
 Manage Identity Providers

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

New Identity Provider

Identity Provider Name*

This name will show as a link on the login page

Identity Provider Description

Allows to authenticate.

Identity Provider Type*

SAML

Identity Provider Metadata

Identity Provider Metadata URL*

[Fetch Metadata](#)

▸ SAML File Metadata (optional)

Advanced SAML Settings

▸ Attribute mappings (optional)

Cancel [Create Identity Provider](#)

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

1. Enter an identity provider name into **Identity Provider Name**.
2. (Optional) Enter a description into **Identity Provider Description**.
3. Specify Identity Provider Metadata from Step 11 of the [Configure Okta as an Identity Provider](#) topic.
 1. Option 1: Enter your **Input Identity Provider Metadata URL** and **Fetch Metadata** to fetch your identity provider metadata from an endpoint.
 2. Option 2: Click **SAML File Metadata (optional)** to upload your metadata XML manually.
4. (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:
 1. Enter a **Group Name**.
 2. 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing

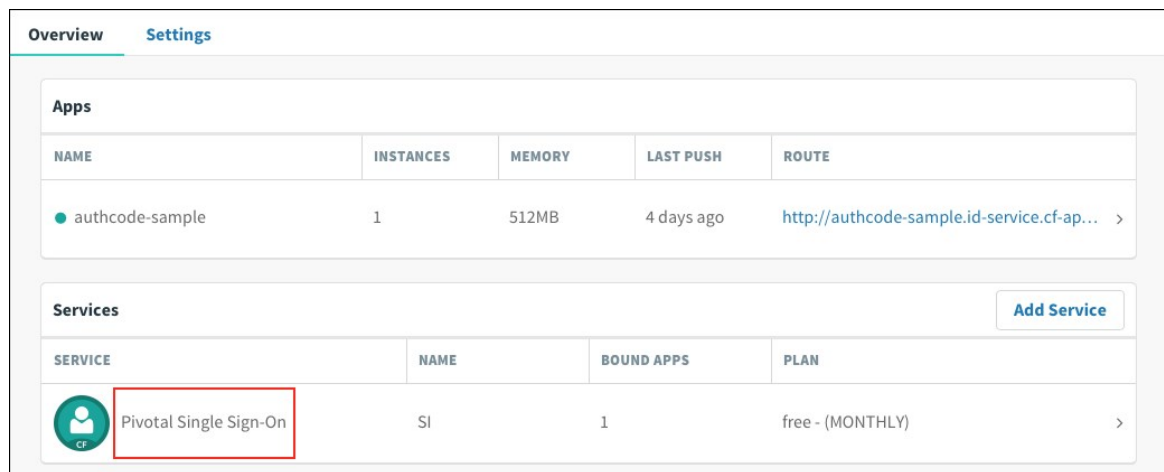


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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**.




Overview **Settings**

Apps

NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap...

Services [Add Service](#)

SERVICE	NAME	BOUND APPS	PLAN
 Pivotal Single Sign-On	SI	1	free - (MONTHLY)



SERVICE **INSTANCE NAME** **SERVICE PLAN**

 **Pivotal Single Sign-On** SI Okta PCF SSO

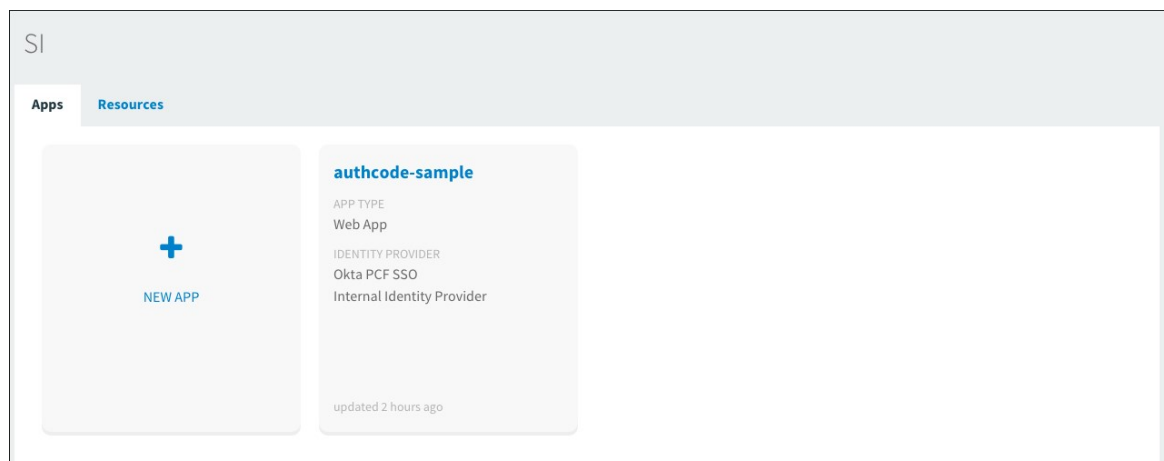
[Manage](#) [Docs](#) [Support](#)

App Binding (1) **Plan** **Settings**

Bound Apps [Edit Bindings](#)

authcode-sample

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



SI

Apps **Resources**

authcode-sample

APP TYPE
Web App

IDENTITY PROVIDER
Okta PCF SSO
Internal Identity Provider

updated 2 hours ago


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


authcode-sample Web App Next Steps

App Name*
authcode-sample

Identity Providers

Select a Identity Provider

 **Okta PCF SSO**

 **Internal User Store**

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X

todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected ▾

Delete
Cancel Save Config

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

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
● authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap... >

- Click the link.

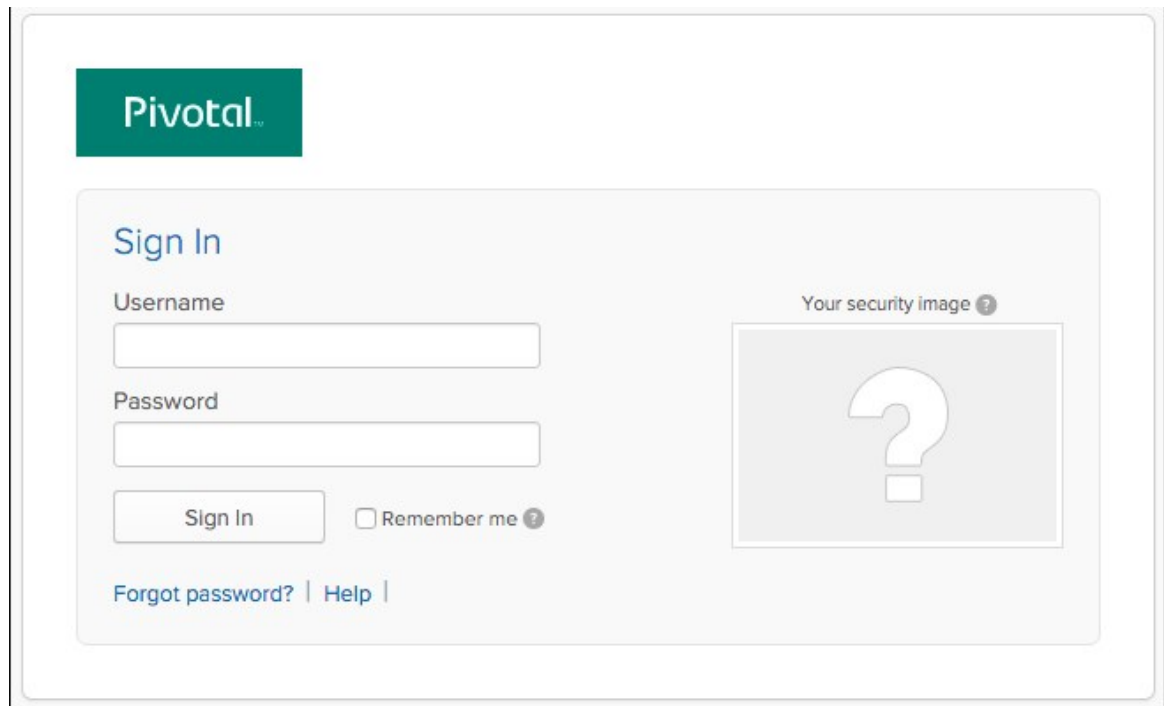
← → ↻ https://authcode-sample

Authcode sample

What do you want to do?

- [Log in via Auth Code Grant Type](#)

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



The image shows a web form for signing in to Pivotal. At the top left is the Pivotal logo. Below it is a 'Sign In' section. This section contains two input fields: 'Username' and 'Password'. Below the password field is a 'Sign In' button and a checkbox labeled 'Remember me' with a help icon. To the right of the input fields is a placeholder for a security image, labeled 'Your security image' with a help icon, showing a large question mark. At the bottom of the sign-in section are two links: 'Forgot password?' and 'Help'.

Pivotal


Sign In

Username

Password

☐ Remember me ?

[Forgot password?](#) | [Help](#) |

Your security image ?


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

Application Authorization

authcode-sample

<https://authcode-sample>

authcode-sample has requested permission to access your account. If you do not recognize this application or its URL, you should click deny. The application will not see your password.

☒ openid

☒ Read TODO list

You can change your approval of permissions or revoke access for this application at any time from account settings. By approving access, you agree to authcode-sample's terms of service and privacy policy.

DENY
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?

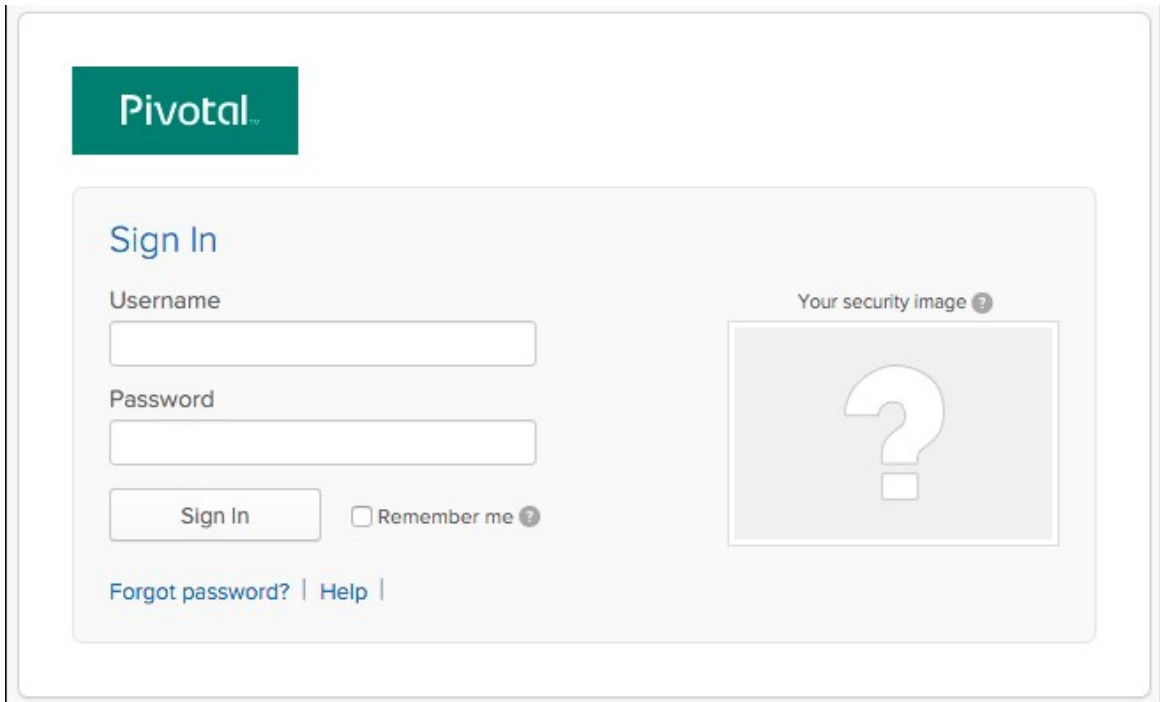
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



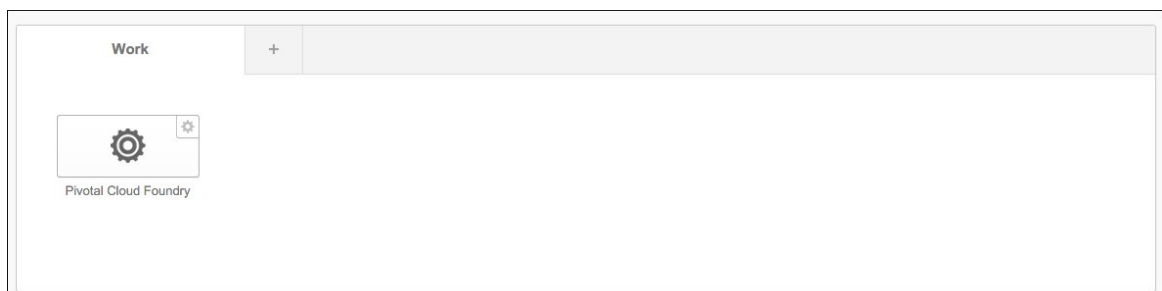
Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign into Okta.

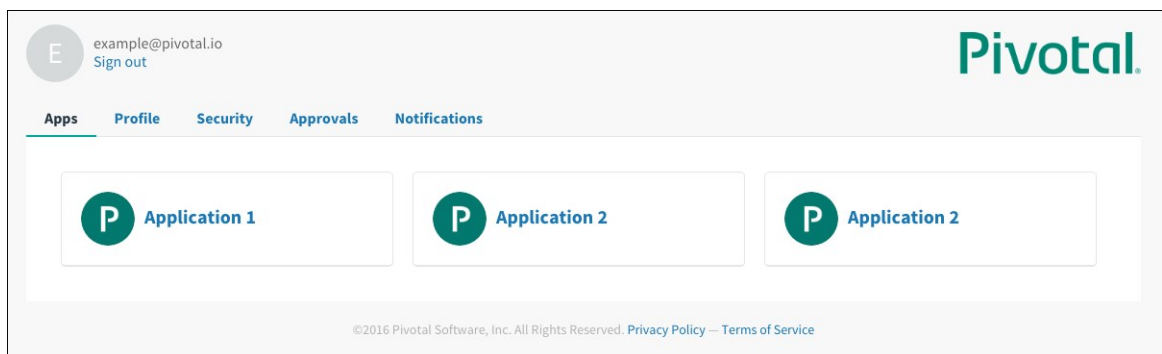


The image shows the Pivotal Sign In page. At the top left is the Pivotal logo. Below it is a 'Sign In' section with two input fields: 'Username' and 'Password'. Below the password field is a 'Sign In' button and a 'Remember me' checkbox with a help icon. To the right of the sign-in fields is a placeholder for a security image, labeled 'Your security image' with a help icon, showing a large question mark. At the bottom of the sign-in section are links for 'Forgot password?' and 'Help'.

- Navigate to the application tile and click it.



- 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.

- Sign into the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.
- Under “What do you want to do?” , click **Log out**.

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](#)

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Troubleshooting

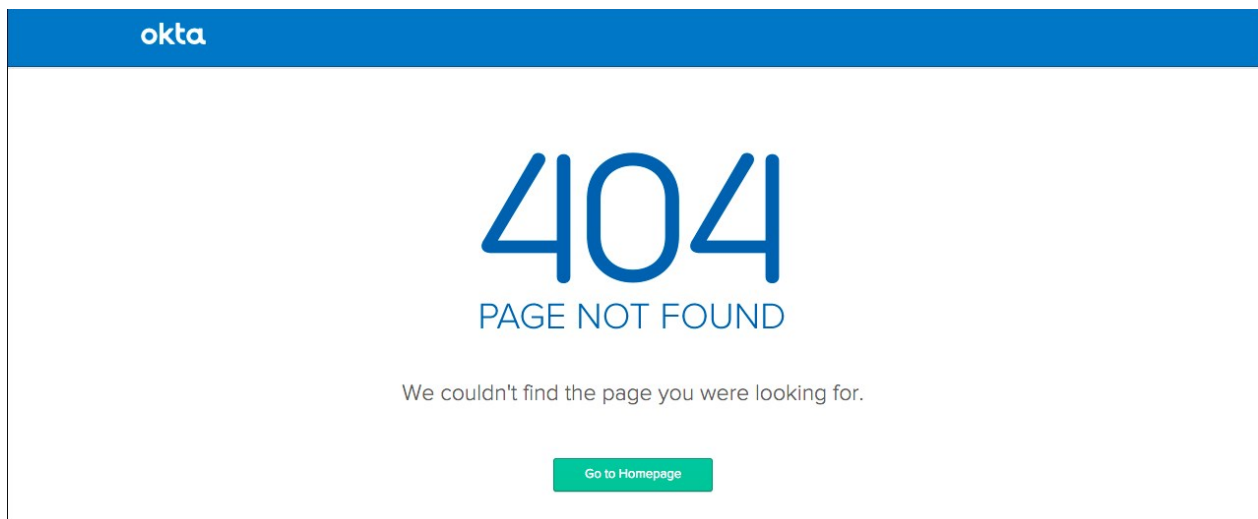


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:

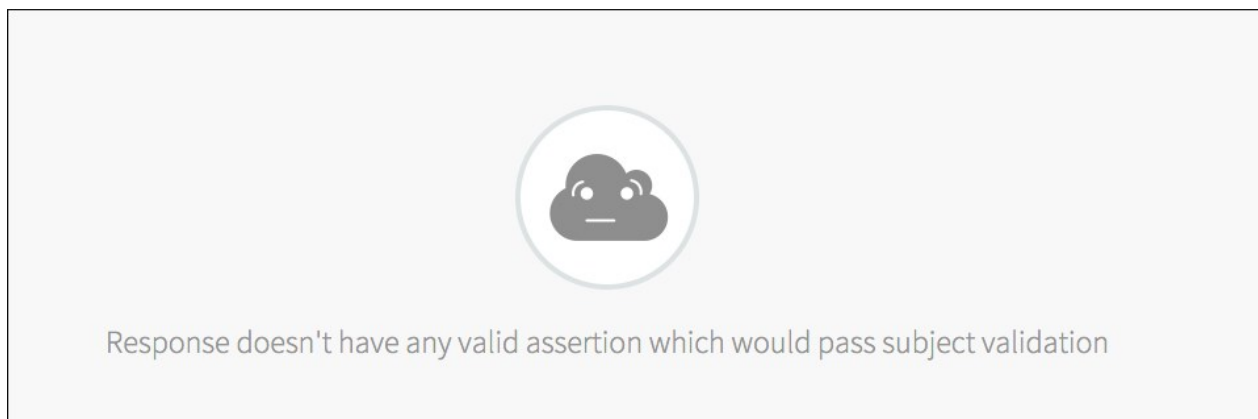


Explanations:

- 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:

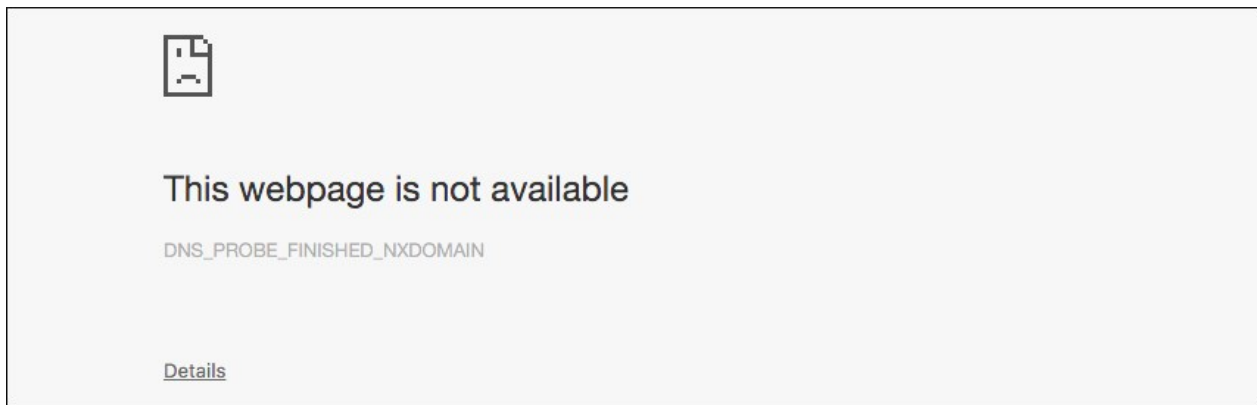


Explanations:

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

Webpage Not Available

Symptom:

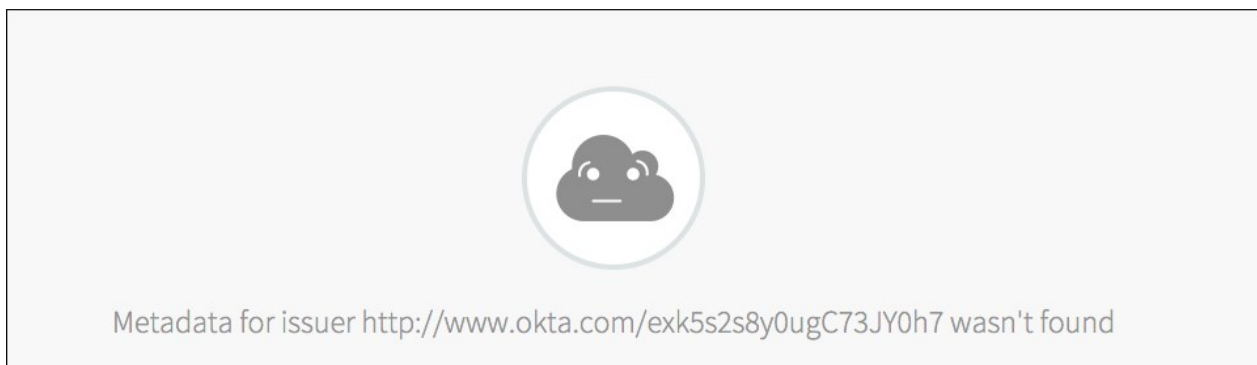


Explanation:

- The SSO URL is misconfigured in Okta.

Metadata Not Found

Symptom:




Explanation:

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

PingFederate Integration Guide

PingFederate Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

Ping

- PingFederate
- A user with admin privileges.



Note: To configure SAML, you must have the Pivotal 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring PingFederate as an Identity Provider

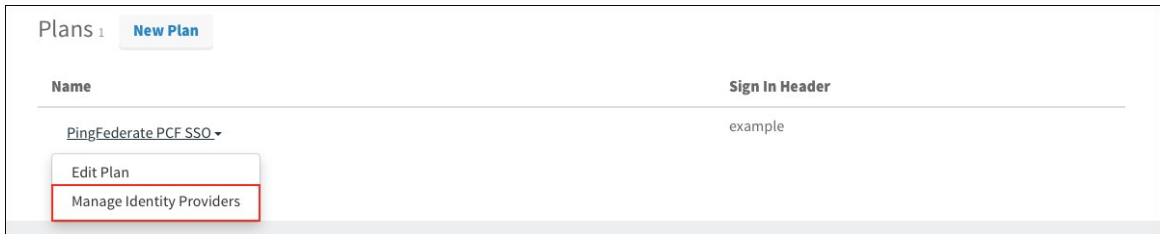


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

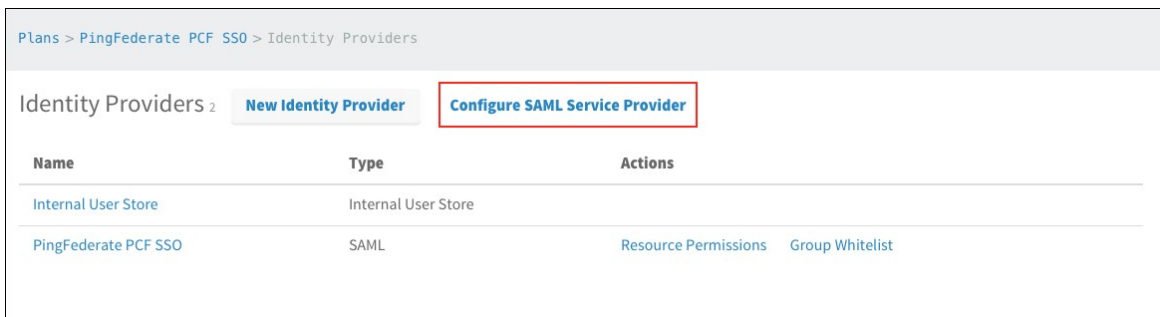
This topic describes how to set up PingFederate as your identity provider by configuring SAML integration in both Pivotal Cloud Foundry (PCF) and PingFederate.

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.

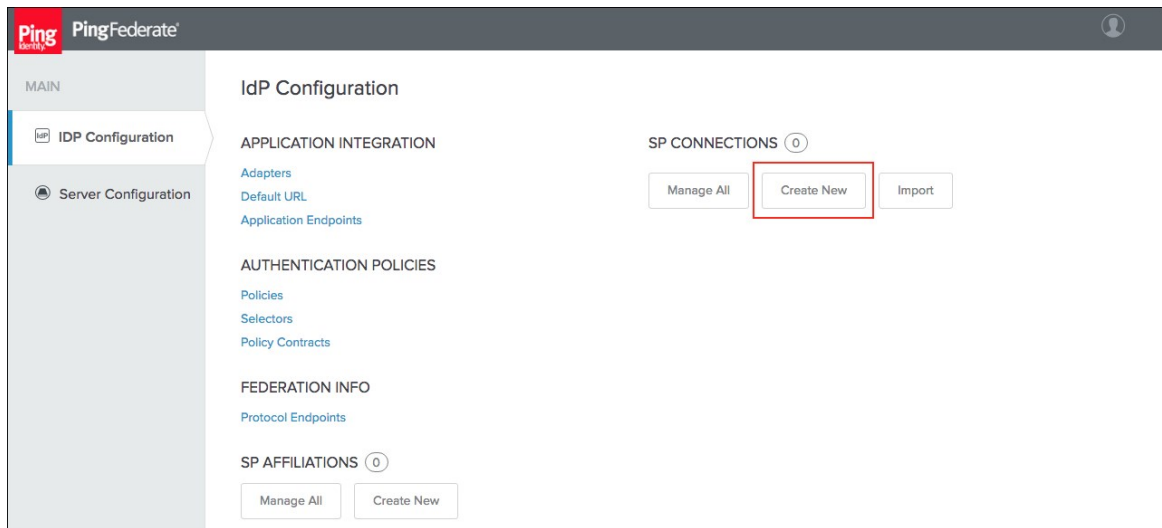


5. (Optional) Select **Require signed assertions** to validate the origin of signed responses.
6. Click **Download Metadata** to download the service provider metadata.
7. Click **Save**.

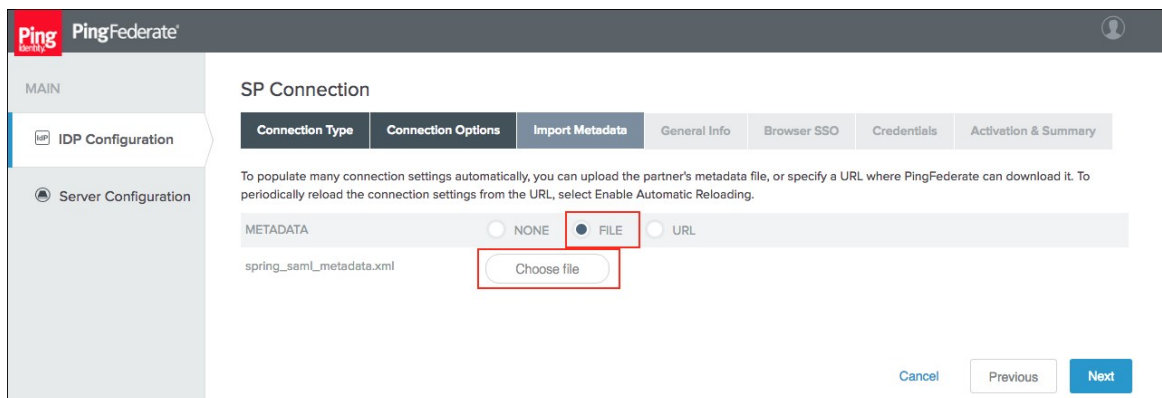
Set up SAML in PingFederate

Configure the Connection

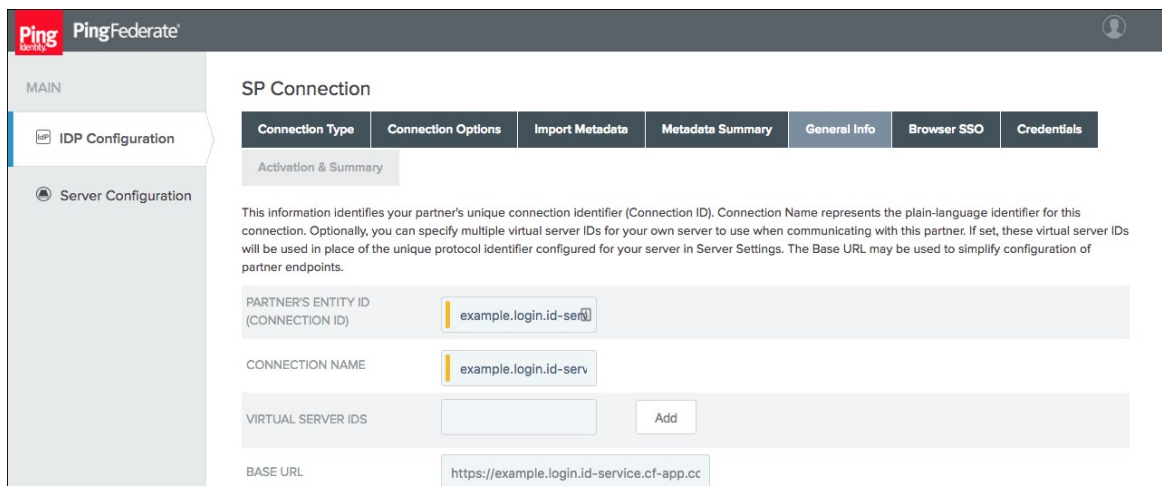
1. Sign in as a PingFederate administrator.
2. Navigate to your identity provider configurations by clicking on the **IDP Configuration** tab.
3. Under **SP Connections**, click the **Create New** button.



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 want 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.

PingFederate

SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance | Mapping Method | Attribute Contract Fulfillment | Issuance Criteria | Summary

Select an IdP adapter instance that may be used to authenticate users for this partner. Attributes returned by the adapter instance you choose (the Adapter Contract) may be used to fulfill the Attribute Contract with your partner.

ADAPTER INSTANCE: Adapter

Adapter Contract

username

☐ OVERRIDE INSTANCE SETTINGS

Manage Adapter Instances

Cancel Save Draft Next

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

PingFederate

SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance | **Mapping Method** | Attribute Contract Fulfillment | Issuance Criteria | Summary

You can choose to fulfill the Attribute Contract with your partner using either the values provided by the "HTTP Basic IdP Adapter" adapter, or you can use these values plus additional attributes retrieved from local data stores.

Adapter Contract

email

givenName

username

☒ USE ONLY THE ADAPTER CONTRACT VALUES IN THE SAML ASSERTION

☐ RETRIEVE ADDITIONAL ATTRIBUTES FROM MULTIPLE DATA STORES USING ONE MAPPING

☐ RETRIEVE ADDITIONAL ATTRIBUTES FROM A DATA STORE -- INCLUDES OPTIONS TO USE ALTERNATE DATA STORES AND/OR A FAILSAFE MAPPING

Cancel Save Draft Previous Next

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

PingFederate

SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance | **Mapping Method** | **Attribute Contract Fulfillment** | Issuance Criteria | Summary

Fulfill your Attribute Contract with values from the authentication adapter or with dynamic text values.

Attribute Contract	Source	Value	Actions
SAML_SUBJECT	Adapter	email	None available

Cancel Save Draft Previous 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**

The screenshot shows the 'Protocol Settings' tab for 'Assertion Consumer Service URL'. The left sidebar has 'IDP Configuration' and 'Server Configuration'. The main content area has tabs for 'Assertion Consumer Service URL', 'Allowable SAML Bindings', 'Artifact Lifetime', 'Artifact Resolver Locations', and 'Signature Policy'. The 'Summary' sub-tab is active, showing a table of SAML bindings. A red box highlights the first row where 'Binding' is 'POST' and 'Endpoint URL' is 'https://example.login.id-service.cf-app.com/saml/SSO/alias/example.login.id-service.cf-app.com'. Below the table is an 'Add' button and a '- SELECT -' dropdown.

Default	Index	Binding	Endpoint URL	Action
default	0	POST	https://example.login.id-service.cf-app.com/saml/SSO/alias/example.login.id-service.cf-app.com	Edit Delete

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

The screenshot shows the 'Allowable SAML Bindings' tab. The left sidebar is the same. The main content area has tabs for 'Assertion Consumer Service URL', 'Allowable SAML Bindings', 'Artifact Resolver Locations', 'Signature Policy', and 'Encryption Policy'. The 'Summary' sub-tab is active, showing a question: 'When the SP sends messages, what SAML bindings do you want to allow?'. There are four radio button options: 'ARTIFACT', 'POST' (which is selected and highlighted with a red box), 'REDIRECT', and 'SOAP'. At the bottom are 'Cancel', 'Save Draft', 'Previous', and 'Next' buttons.

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**.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring a Single Sign-On Service Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

Plans 1 [New Plan](#)

Name	Sign In Header
PingOne PCF SSO	example

[Edit Plan](#)

[Manage Identity Providers](#)

3. Click **New Identity Provider**.

New Identity Provider

Identity Provider Name*

This name will show as a link on the login page

Identity Provider Description

Allows to authenticate.

Identity Provider Type*

SAML

Identity Provider Metadata

Identity Provider Metadata URL*

[Fetch Metadata](#)

▶ SAML File Metadata (optional)

Advanced SAML Settings

▶ Attribute mappings (optional)

[Cancel](#) [Create Identity Provider](#)

4. To create a new identity provider, perform the following steps:
1. Enter an identity provider name into **Identity Provider Name**.
 2. (Optional) Enter a description into **Identity Provider Description**.
 3. Click **SAML File Metadata (optional)**, then click the **Upload Identity Provider Metadata** button to upload your metadata XML.
 4. (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:
1. Enter a **Group Name**.
 2. 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.

- Click **Group Whitelist** and enter the group names from the external identity provider to propagate in the ID token when a user authenticates.

Create a pull request or raise an issue on the source for this page in [GitHub](#)

Testing




Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

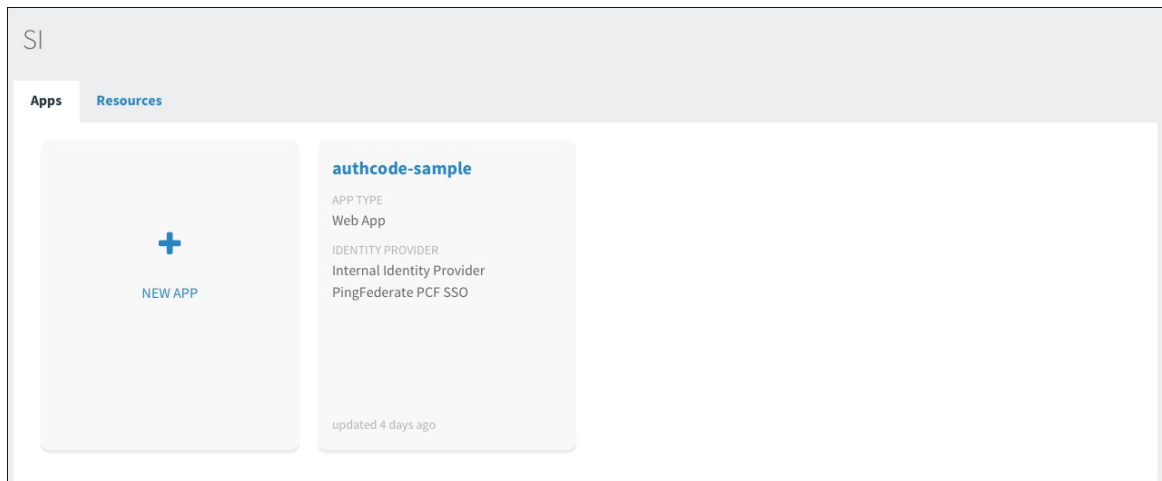
- Log in to Apps Manager at <https://apps.YOUR-SYSTEM-DOMAIN> and navigate to the organization and space where your application is located.
- Under **Services**, locate the service instance of the Single Sign-On (SSO) plan bound to your application. Click the service instance and then click **Manage**.

The screenshot shows the 'Services' tab in the Apps Manager interface. It features a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. The 'Pivotal Single Sign-On' service is highlighted with a red box. The table also includes an 'Add Service' button.

SERVICE	NAME	BOUND APPS	PLAN
 Pivotal Single Sign-On	SI	1	free - (MONTHLY)

The screenshot shows the 'Pivotal Single Sign-On' service instance page. It includes a 'Manage' link highlighted with a red box, along with 'Docs' and 'Support' links. The page also shows the 'App Binding (1)' tab and a 'Bound Apps' section with the application 'authcode-sample' listed.

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



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

authcode-sample Web App Next Steps

App Name*

authcode-sample

Identity Providers

Select a Identity Provider

Internal User Store PingFederate PCF SSO

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected

Delete Cancel Save Config

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

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap... >

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



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

A 'Sign On' form is displayed within a light gray box. At the top, the text 'Sign On' is in a large, bold font. Below it are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both labels are in a light gray font. At the bottom right of the form is a blue button with the white text 'Login'.

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

Application Authorization

authcode-sample

<https://authcode-sample>

authcode-sample has requested permission to access your account. If you do not recognize this application or its URL, you should click deny. The application will not see your password.

☒ openid

☒ Read TODO list

You can change your approval of permissions or revoke access for this application at any time from account settings. By approving access, you agree to authcode-sample's terms of service and privacy policy.

DENY
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" ],
  "token_type" : "bearer",
  "expires_in" : 3600
}
```



```

"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" : "all12e6d9-8a23-47be-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" : "all12e6d9-8a23-47be-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" : "all12e6d9-8a23-47be-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.

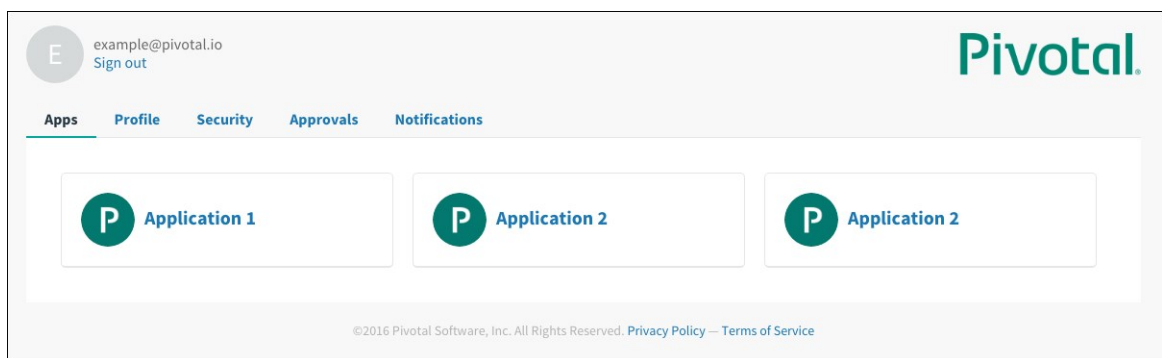
Sign On

Username

Password

Login

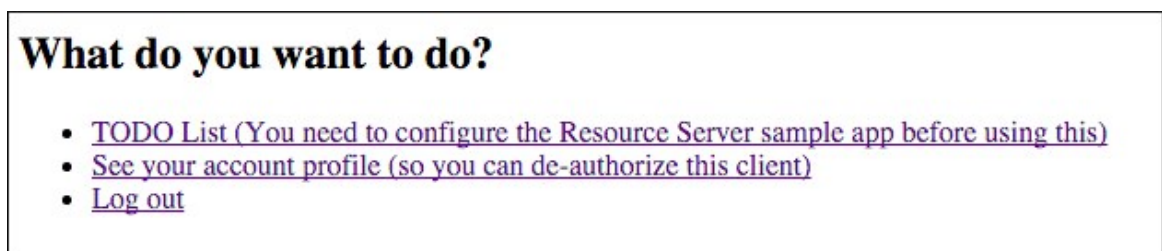
2. Navigate to your application and click it.
3. View the list of applications you have access to.



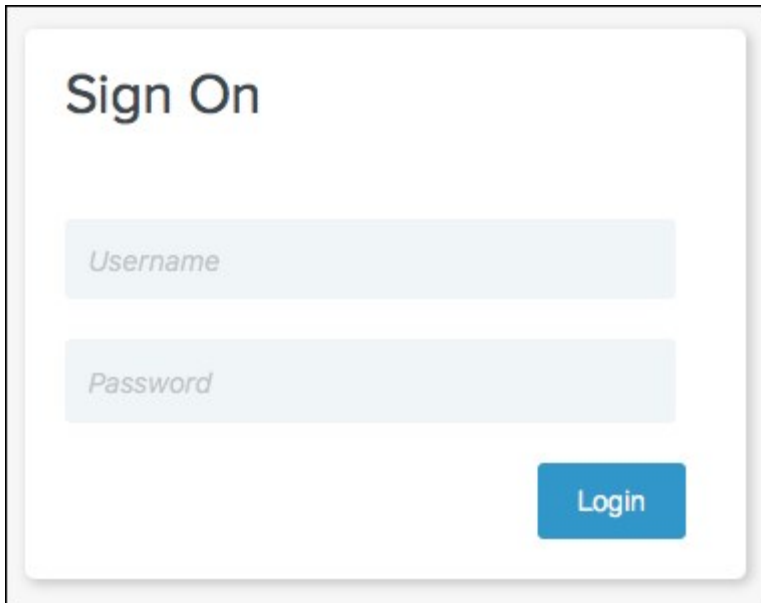
Test Your Single Sign-Off

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.

A screenshot of a 'Sign On' form. The form has a title 'Sign On' at the top. Below the title are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both fields are light blue with placeholder text. Below the 'Password' field is a blue button with the text 'Login' in white.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Troubleshooting

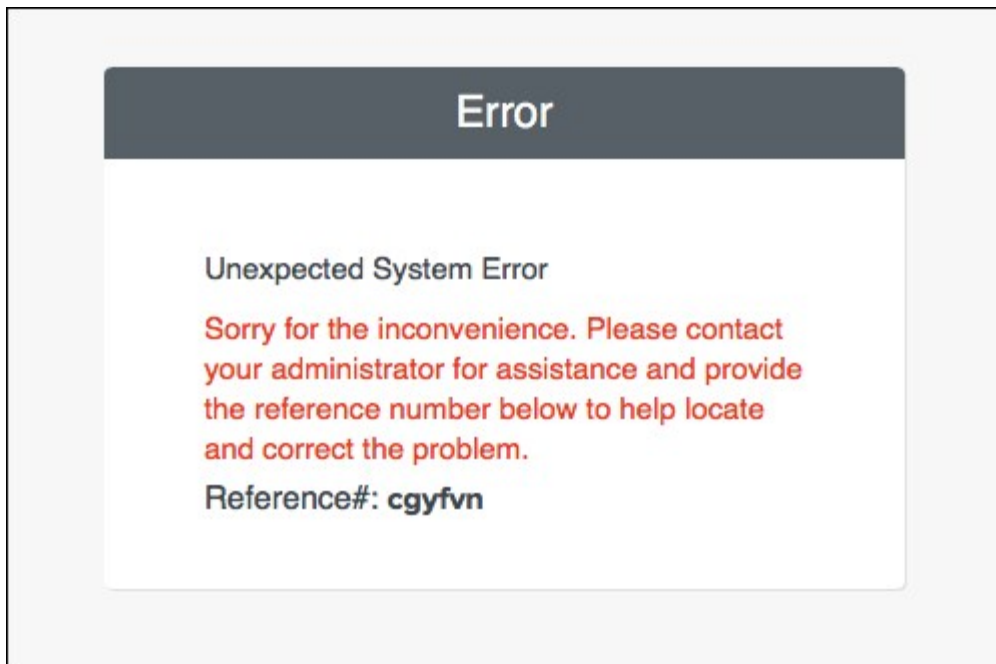


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:

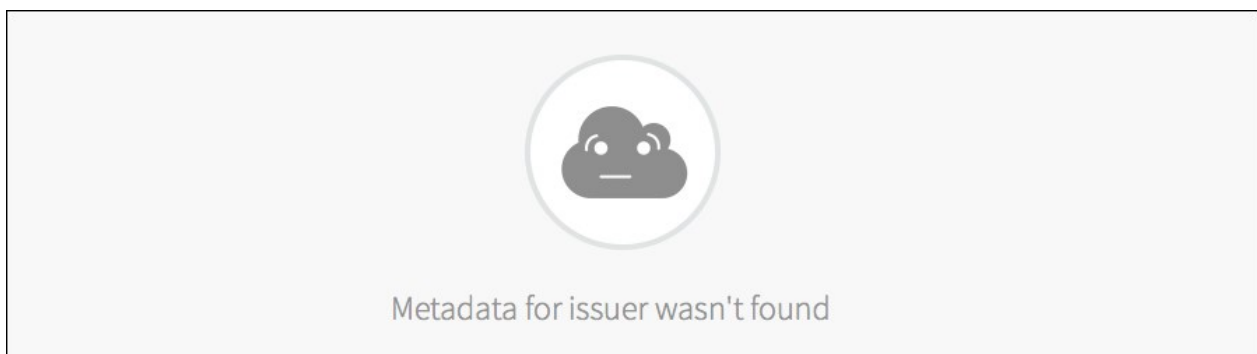


Explanations:

- 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

PingOne Cloud Integration Guide

PingOne Cloud Integration Guide Overview



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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, v1.7.0 or later.
- Single Sign-On, v1.1.0 or later.

PingOne Cloud

- PingOne Cloud
- A user with Application Admin privileges.



Note: To configure SAML, you must have the Pivotal 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](#)

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring PingOne Cloud as an Identity Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

The screenshot shows the SSO dashboard with a 'Plans' tab selected. A dropdown menu is open for the 'PingOne PCF SSO' plan, and the 'Manage Identity Providers' option is highlighted with a red rectangle.

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

The screenshot shows the 'Identity Providers' page. The 'Configure SAML Service Provider' button is highlighted with a red rectangle. Below the button is a table with two rows: 'Internal User Store' and 'PingOne PCF SSO'.

Name	Type	Actions
Internal User Store	Internal User Store	
PingOne PCF SSO	SAML	Resource Permissions Group Whitelist

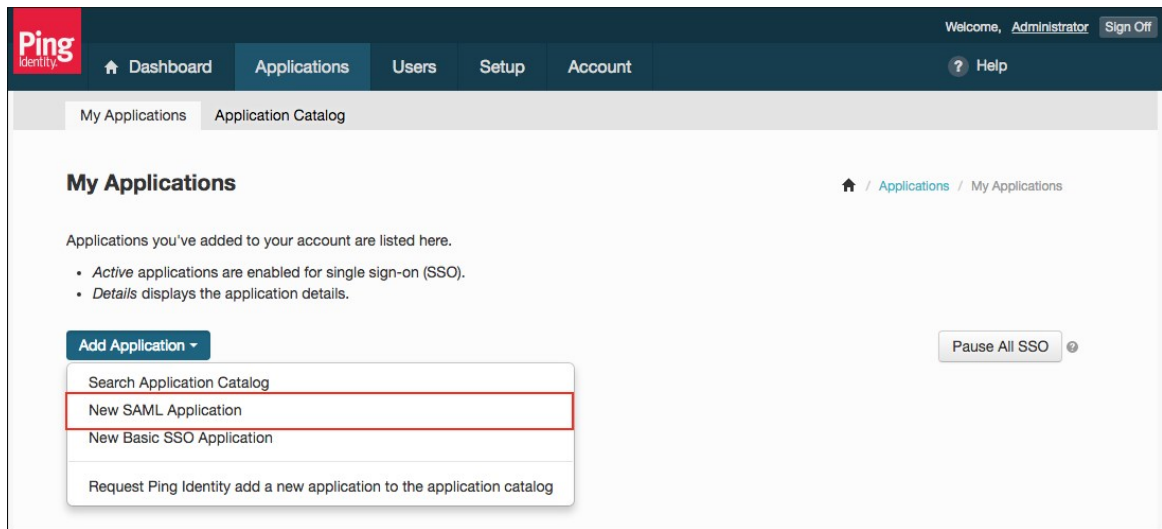
4. (Optional) Select **Perform signed authentication requests** to enforce SSO private key signature and identity provider validation.

The screenshot shows the 'Configure SAML Service Provider' form. The 'Perform signed authentication requests' checkbox is checked. There is also a 'Require signed assertions' checkbox which is unchecked. A 'Save' button is at the bottom left, and a 'Download Metadata' button is at the top right.

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.
-

2. Application Configuration

I have the SAML configuration
I have the SSO URL

You will need to download this SAML metadata to configure the application:

SAML Metadata [Download](#)

Provide SAML details about the application you are connecting to:

Protocol Version ☒ SAML v 2.0 ☐ SAML v 1.1

Upload Metadata ⓘ Select File [Or use URL](#)

Assertion Consumer Service (ACS)

Entity ID

Application URL

Single Logout Endpoint ⓘ

Single Logout Response Endpoint ⓘ

Single Logout Binding Type ☐ Redirect ☒ Post

Verification Certificate ⓘ Choose File No file chosen
saml20metadata.cer

Signing Algorithm

Force Re-authentication ⓘ ☐

Keep the following in mind when creating your connection:

1. Both SP- and IdP-Initiated SSO are allowed
2. Map SAML_SUBJECT in your attribute contract, plus any attributes (configure them in PingOne later)
3. Allow outbound POST or redirect bindings
4. Allow inbound POST

NEXT: SSO Attribute Mapping

Cancel
Back
Continue to Next Step

6. In the **Application Configuration** section, perform the following steps:
 1. Select **I have the SAML configuration**.
 2. For **SAML Metadata**, click **Download** to download the identity provider metadata.
 3. For **Protocol Version**, select **SAML v 2.0**.
 4. For **Upload Metadata**, click **Select File** and select the service provider metadata.
 5. 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.

3. SSO Attribute Mapping

Map the necessary application provider (AP) attributes to attributes used by your identity provider (IdP).

	Application Attribute	Identity Bridge Attribute or Literal Value	Required
1	firstName	First Name <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
2	lastName	Last Name <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
3	email	Email <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
4	group	memberOf <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>

NEXT: Review Setup

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

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring a Single Sign-On Service Provider



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Setting up SAML

- Log into the SSO dashboard at <https://p-identity.YOUR-SYSTEM-DOMAIN> as a Plan Administrator.
- Select your plan and click **Manage Identity Providers** on the drop-down menu.

Plans 1

Name	Sign In Header
PingOne PCF SSO ▾	example
<input type="button" value="Edit Plan"/> <input type="button" value="Manage Identity Providers"/>	

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

New Identity Provider

Identity Provider Name*

This name will show as a link on the login page

Identity Provider Description

Allows Enter a group name to authenticate.

Identity Provider Type*

SAML

Identity Provider Metadata

Identity Provider Metadata URL*

[Fetch Metadata](#)

▸ SAML File Metadata (optional)

Advanced SAML Settings

▸ Attribute mappings (optional)

[Cancel](#) [Create Identity Provider](#)

4. To create a new identity provider, perform the following steps:
 1. Enter an identity provider name into **Identity Provider Name**.
 2. (Optional) Enter a description into **Identity Provider Description**.
 3. Click **SAML File Metadata (optional)** followed by clicking the **Upload Identity Provider Metadata** button to upload your metadata XML.
 4. (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:
 1. Enter a **Group Name**.
 2. 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.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing



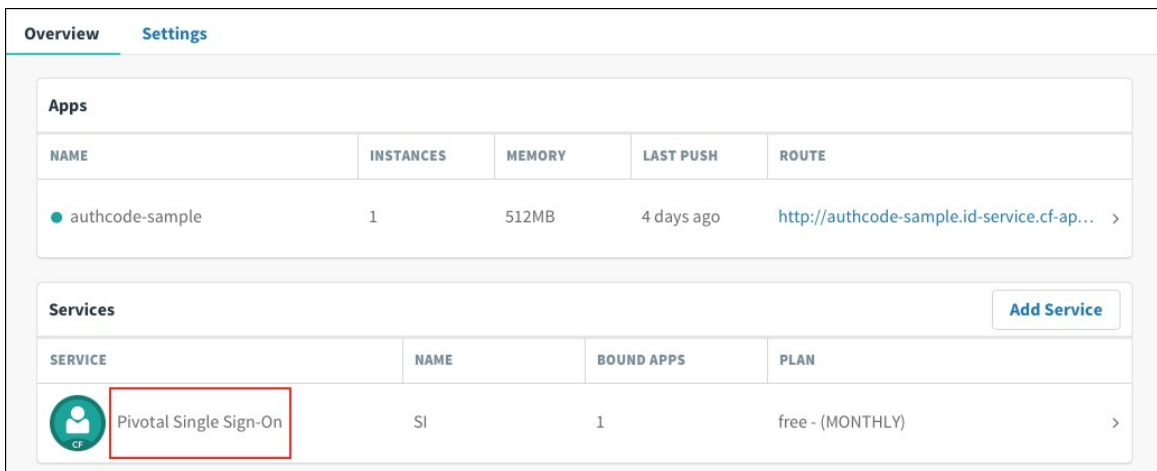
Warning: Single Sign-On for PCF v1.6 is no longer supported because it has

reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.


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**.



The screenshot shows the 'Services' tab in the Apps Manager interface. It features a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. The 'Pivotal Single Sign-On' service is listed with a name of 'SI' and a plan of 'free - (MONTHLY)'. A red box highlights the service name 'Pivotal Single Sign-On'.

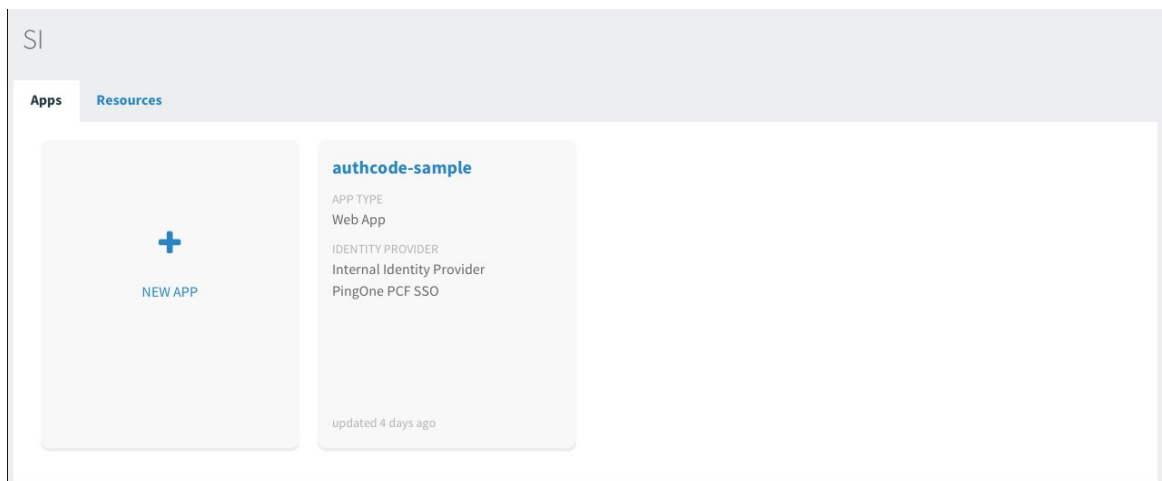
SERVICE	NAME	BOUND APPS	PLAN
 Pivotal Single Sign-On	SI	1	free - (MONTHLY)



The screenshot shows the details page for the 'Pivotal Single Sign-On' service. It includes tabs for 'App Binding (1)', 'Plan', and 'Settings'. The 'Manage' button is highlighted with a red box. Below the tabs, there is a 'Bound Apps' section showing the application 'authcode-sample'.

SERVICE	INSTANCE NAME	SERVICE PLAN
 Pivotal Single Sign-On	SI	PingOne PCF SSO

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



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

authcode-sample Web App Next Steps

App Name*

authcode-sample

Identity Providers

Select a Identity Provider

Internal User Store PingOne PCF SSO

Redirect URIs

The Authentication Response will be sent to the following locations:

Auth Redirect URIs*
Provide a comma-separated list of URIs

https://authcode-sample.id-service.cf-app.com

Authorization

Scopes
Permissions requested by the application

todo

todo.read X todo.write X

System Provided

openid X

Select Scopes

Auto-Approved Scopes
Permissions automatically approved on behalf of the user

None selected ▾

Delete Cancel Save Config

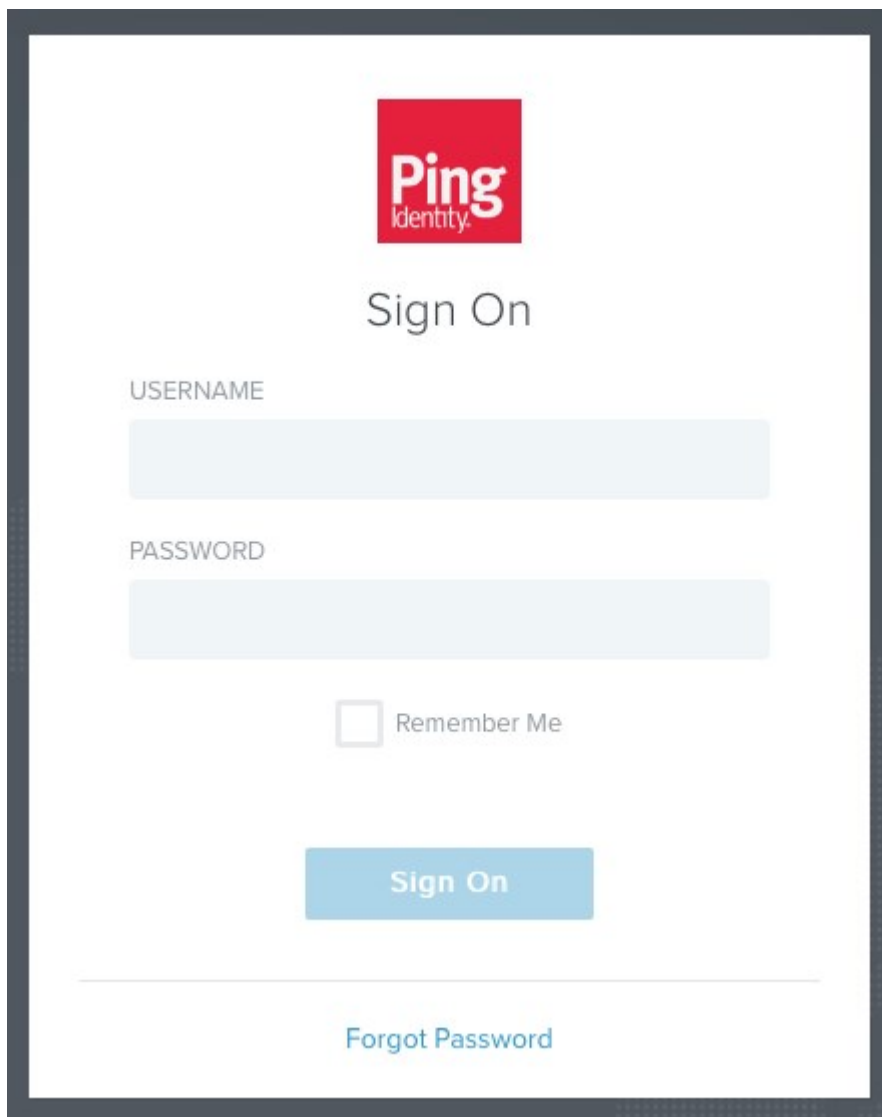
- Return to Apps Manager and click on the URL below your application to be redirected to the identity provider to authenticate.

Overview Settings				
Apps				
NAME	INSTANCES	MEMORY	LAST PUSH	ROUTE
authcode-sample	1	512MB	4 days ago	http://authcode-sample.id-service.cf-ap... >

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**.

Application Authorization

authcode-sample

<https://authcode-sample>

authcode-sample has requested permission to access your account. If you do not recognize this application or its URL, you should click deny. The application will not see your password.

☒ openid

☒ Read TODO list

You can change your approval of permissions or revoke access for this application at any time from account settings. By approving access, you agree to authcode-sample's terms of service and privacy policy.

DENY
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" ]
}
```

```

"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?

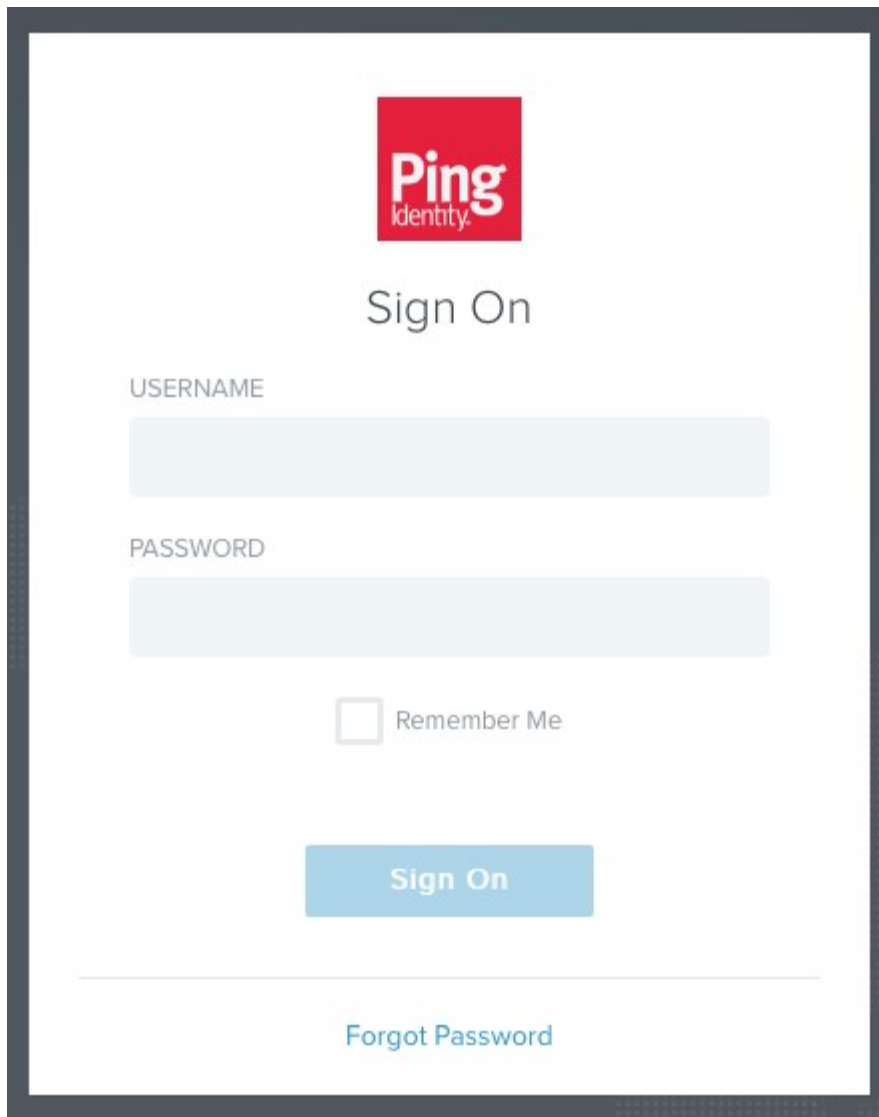
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection

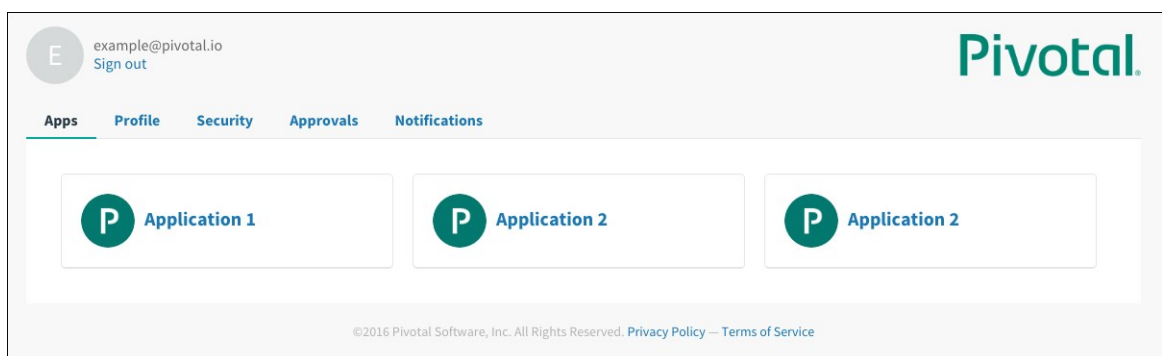


Note: SSO does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select applications assigned to the user.

1. Sign in to PingOne.

The image shows a Ping Identity 'Sign On' page. At the top is the Ping Identity logo, which consists of a red square with the word 'Ping' in white and 'Identity.' in smaller white text below it. Below the logo is the text 'Sign On' in a large, dark font. Underneath is a form with two input fields: 'USERNAME' and 'PASSWORD', each with a light blue border. Below the password field is a checkbox labeled 'Remember Me'. A blue 'Sign On' button is centered below the form. At the bottom, there is a horizontal line and a link that says 'Forgot Password' in blue text.

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.

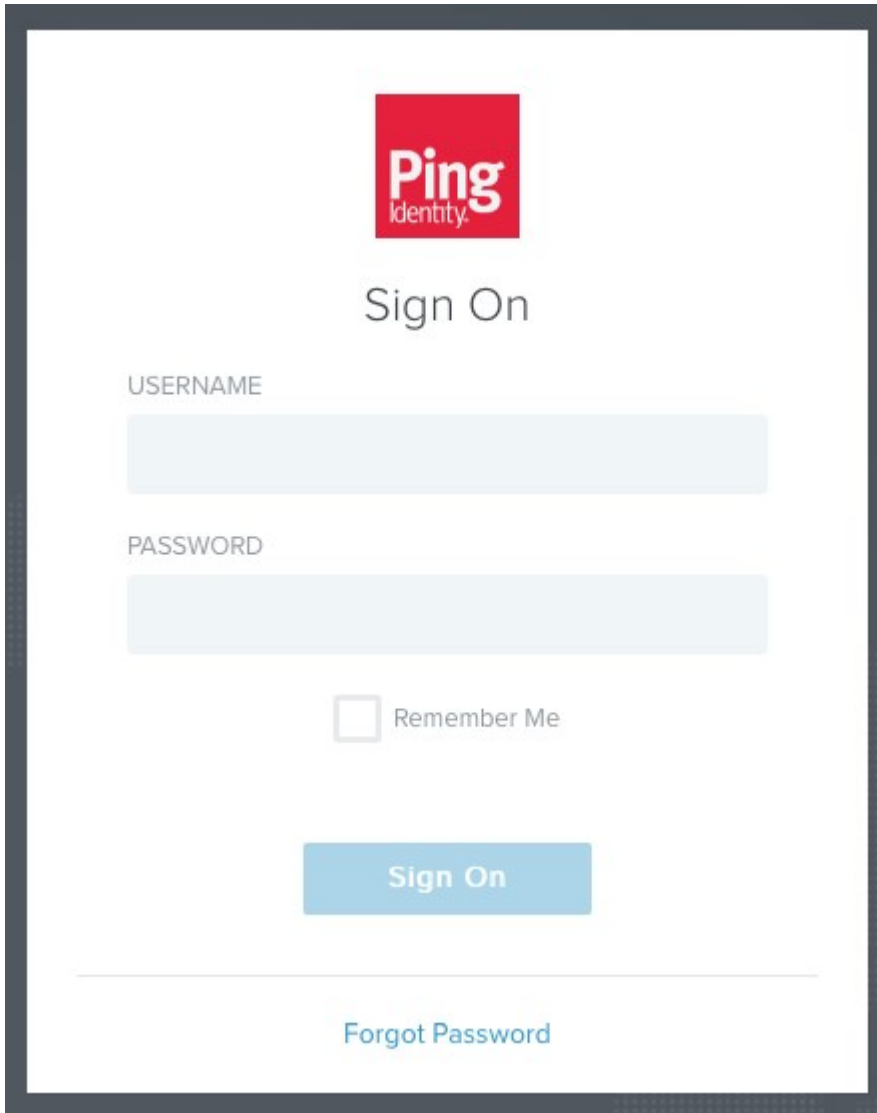
1. Sign into the sample application. Information about the access and ID token displays, as well as the “What do you want to do?” section.

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

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](#)

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

The image shows the Ping Identity Sign On page. At the top is the Ping Identity logo, which consists of a red square with the word "Ping" in white and "Identity." in smaller white text below it. Below the logo is the text "Sign On". Underneath is a form with two input fields: "USERNAME" and "PASSWORD". Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a blue button labeled "Sign On". Below the button is a horizontal line, and below that is a link labeled "Forgot Password".

Create a pull request or raise an issue on the source for this page in GitHub

Troubleshooting

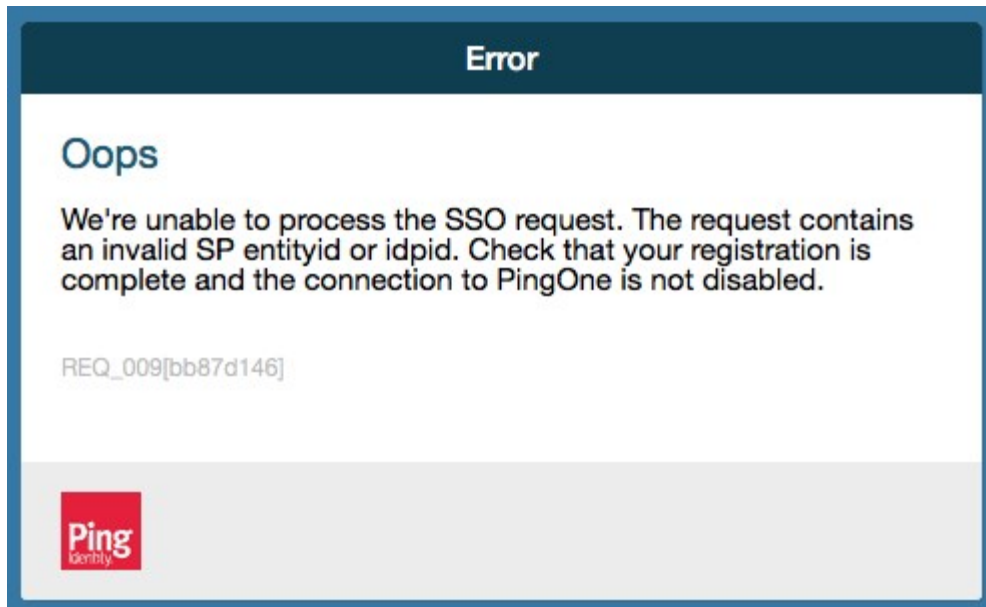


Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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:

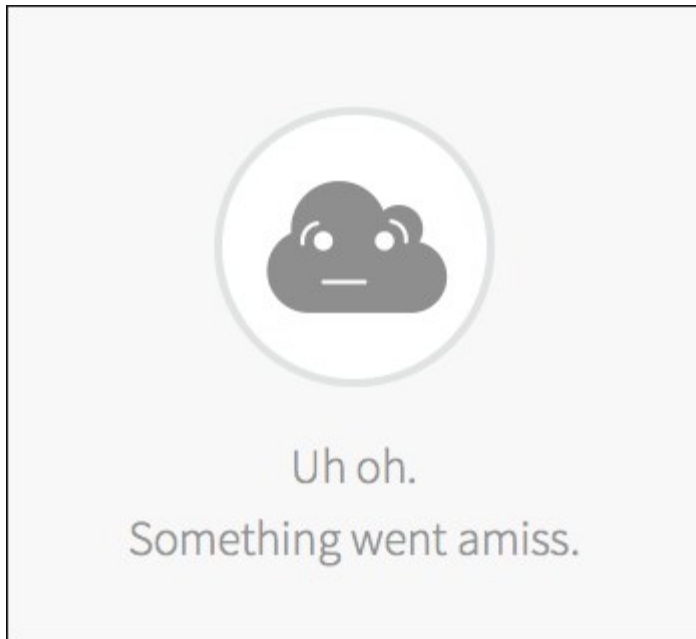


Explanations:

- 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:

Identity Provider Metadata

Identity Provider Metadata URL *

Fetch Metadata

Error processing metadata

▼ SAML File Metadata (optional)

Upload Identity Provider Metadata

Explanation:

- The identity provider metadata is missing configurations for Name ID. See [Configure Identity Provider Metadata](#).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Plan-to-Plan OIDC Integration Guide

Plan-to-Plan OIDC Integration Guide



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how to set up the Pivotal Cloud Foundry (PCF) Single Sign-On (SSO) to integrate a SSO Service Plan as an OpenID Connect (OIDC) identity provider.

Service plans are represented in User Access and Administration (UAA) as identity zones. UAA provides the ability to integrate any two UAAs with one acting as the relying party and the other acting as the identity provider. This includes identity zones within the same multi-tenant UAA, as well as separate UAA instances, such as the Bosh UAA, Ops Manager UAA, or a standalone UAA (provided they are on a version that has OIDC implemented). This topic explains how you can perform the integration from one SSO service plan to another through the SSO service tile.

Prerequisites

To integrate Plan-to-Plan OIDC with PCF, you need:

- PCF, v1.12 or later
- Single Sign-On, v1.5.0 or later
- An active SSO Service Plan that will act as an identity provider
- A second active SSO Service Plan that will act as the relying party
- A user with admin privileges



Note: To configure OIDC according to these steps, you must have the Single Sign-

On service broker installed in your PCF deployment. You need to create a plan, add any plan administrators, and specify any organizations for which this plan should be the authentication authority. For help configuring plans, see [Managing Service Plans](#).

Integrating a Plan-to-Plan OIDC for SSO

Complete this process to set up Plan-to-Plan OIDC integration for the SSO service. For more information, see [Configuring Plan-to-Plan OIDC Integrations](#).

Testing the OIDC Connection

After you have configured the Plan-to-Plan OIDC integration for SSO, you can test it to confirm it works. For more information, see [Testing](#).

Troubleshooting

For information about common configuration problems and error states, see [Troubleshooting](#).

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Configuring Plan-to-Plan OIDC Integration



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how to set up the Plan-to-Plan OpenID Connect (OIDC) integration between two Single Sign-On service plans, one acting as an identity provider (“identity provider plan” or IDP) and one acting as a relying party (“relying party plan” or RP).

Doing this allows users from the identity provider plan to authenticate into the relying party plan through OIDC.

Setting Up Relying Party Configurations in the Identity Provider Plan

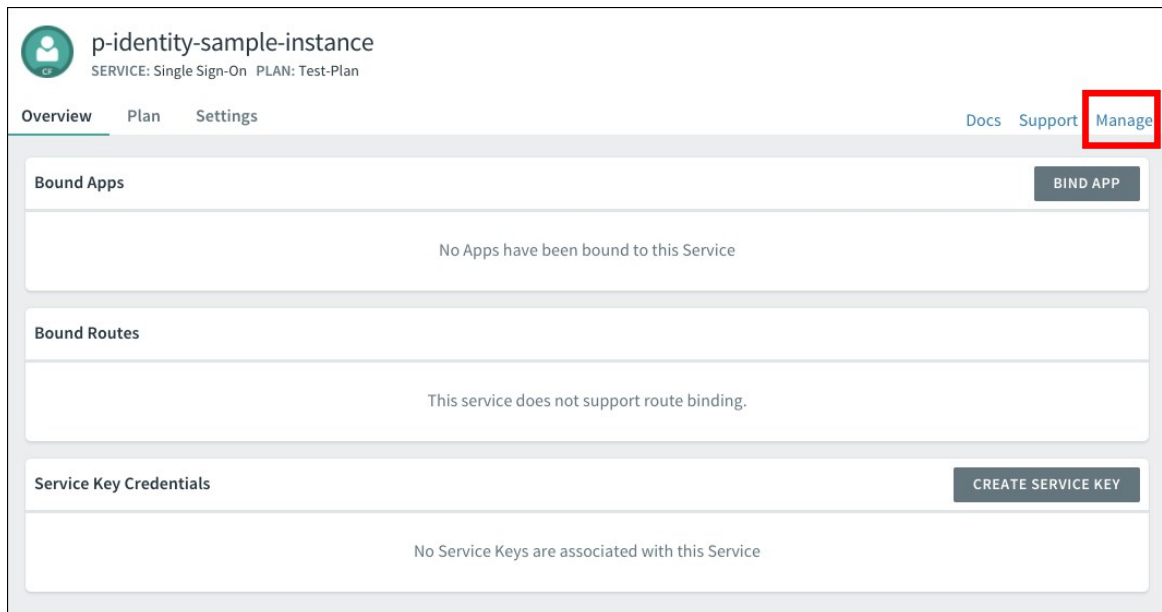
Prerequisites

- Your IDP must be visible to your Org.
- You must add the IDP as a service instance in a Space so you can access the app developer dashboard.

If you haven't completed these prerequisites, see [Create or Edit Service Plans](#).

1. Navigate to Apps Manager.
2. Select the Space.
3. Click into the **Service** tab.

4. Click to select the service you want to modify.
5. Click **Manage**.



6. Click **New App**. The New App page appears.
7. Type a name in the **App Name** field.
8. Choose **Web App** from the list of Application Types.
9. Type a temporary URL in the **Auth Redirect URIs** field. You'll replace this URL when you have configured an identity provider on the relying party plan.
10. In the **Scopes** field, type `openid`.
Optionally, select `openid` from the list of **Auto-Approved Scopes**. By adding `openid` as an automatically approved scope, you will keep users from being prompted to authorize a login from the identity provider.
11. Click **Create App**. If the app is created successfully, you will be prompted to download your app credentials.

Download App Credentials ✕

App ID

App Secret

[Show App Secret](#)

This is the last time these App Credentials will be available for download.

Download App Credentials

- Click **Download App Credentials** to save the credentials for your application.



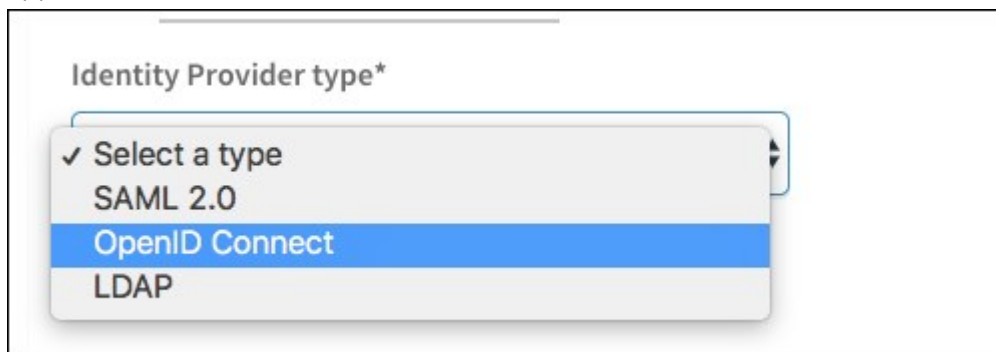
Important: This is the last time you will be able to download your app credentials. Pivotal strongly recommends that you download the credentials and store them securely.

Setting Up the OIDC Identity Provider Configuration in the Relying Party Plan

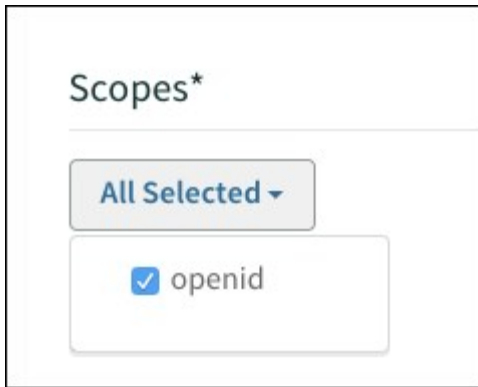
- Navigate to <https://p-identity.YOUR-SYSTEM-DOMAIN>.
- Log into the SSO dashboard using the credentials associated with your UAA administrator account. You can find these credentials in your Pivotal Application Service tile in Ops Manager. For more information, see [Logging in to Apps Manager](#).
- Click the Relying Party plan name and choose **Manage Identity Providers** from the dropdown.



4. Click **New Identity Provider**. The New Identity Provider screen appears.
5. Enter an **Identity Provider Name**. This string, in lowercase with dashes replacing spaces, will become your Origin Key. For example, “My Test Provider” will become “my-test-provider.”
6. Enter a **Description**. This description will be visible to Space developers when they select an IDP for their application.
7. Select **OpenID Connect** as the **Identity Provider type**. The OpenID Connect Settings appear.



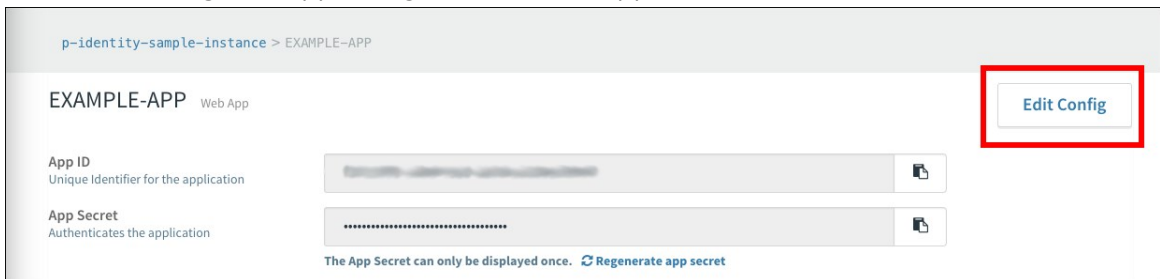
8. If you're using a self-signed certificate for PCF where the IDP is located, select the **Skip SSL Validation** checkbox. If you're not using a self-signed certificate, you can leave this box unchecked.
9. Select the **Enable Discovery** checkbox and type in the **Discovery Endpoint URL**. This URL will be `https://IDP_AUTH_DOMAIN/.well-known/openid-configuration`, where `IDP_AUTH_DOMAIN` is the Auth Domain setting you entered when you created the IDP service plan you are integrating with.
10. Fill in the **Relying Party OAuth Client ID** with the App Client ID from [the previous section](#).
11. Fill in the **Relying Party OAuth Client Secret** with the App Secret from [the previous section](#).
12. Confirm that `openid` is selected as a Scope by clicking **All Selected**.



Finalizing Configuration

Once you've created an app, you can return to the App page to finish configuration.

1. Return to the [app you created](#).
2. Click **Edit Config**. The app configuration screen appears.



3. Add a **Auth Redirect URL**. The URL should read `https://RP_AUTH_DOMAIN/login/callback/ORIGIN_KEY`, where the `RP_AUTH_DOMAIN` is the Auth Domain setting you entered during RP configuration and the `ORIGIN_KEY` is based on the IDP name you set in the SSO dashboard.
4. Click **Save Config**.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

Testing OIDC Integrations



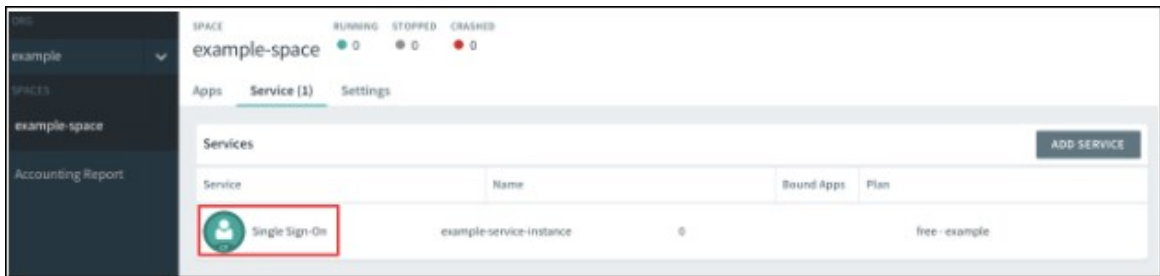
Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how a Pivotal Cloud Foundry (PCF) administrator can test the OpenID Connect (OIDC) connection between a Single Sign-On (SSO) service plan acting as an Identity Provider (IDP), and another SSO service plan acting as a Relying Party (RP).

Testing Your SSO Connection

1. Log in to Apps Manager at `https://apps.YOUR-SYSTEM-DOMAIN`.

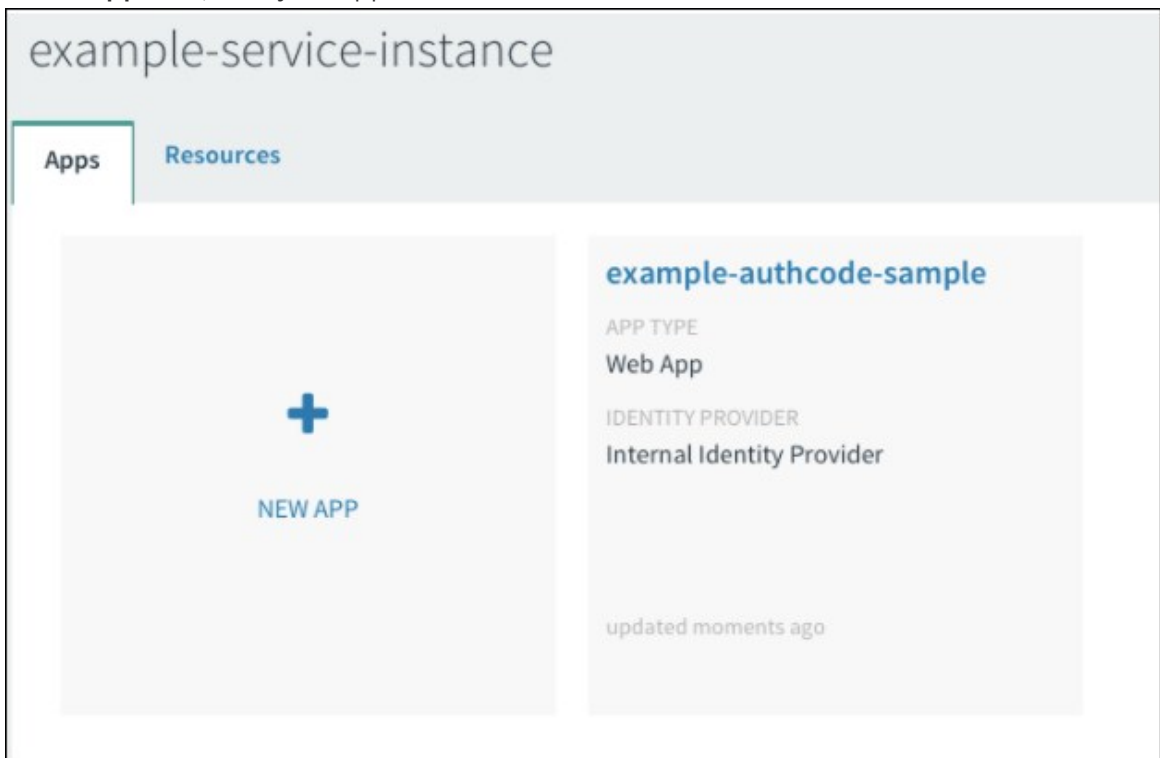
- Navigate to the Org and Space where your application is located.
- Locate the service instance of the SSO plan bound to your app.



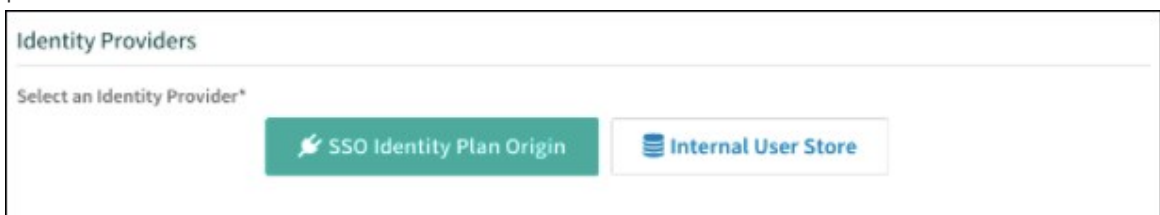
- Select the service instance.
- Click **Manage**.



- In the **Apps** tab, click your app.

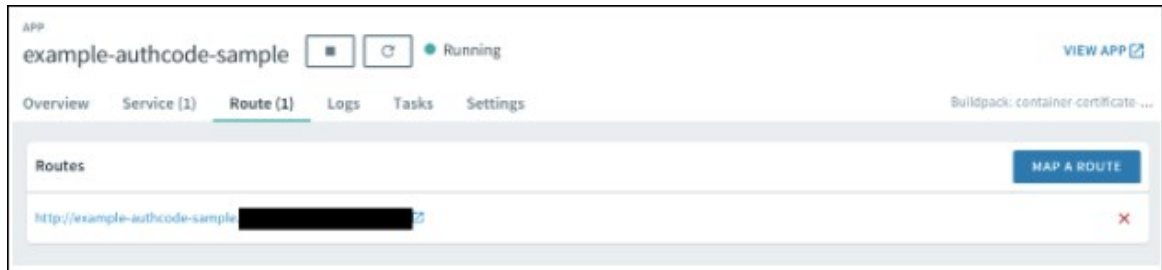


- Under **Identity Providers**, select the SSO Identity Plan IDP. Remove any other identity providers.



- Return to Apps Manager.

- Click the URL listed below your app to access the application.



- Log in to the app. You will be redirected to the IDP to authenticate.
- Sign in to the IDP.
- If necessary, authorize the necessary scopes to connect the IDP with your app. If you need to do this, the IDP will prompt you.
- After authorizing the scopes, you should be logged into the app.

[Create a pull request or raise an issue on the source for this page in GitHub](#)

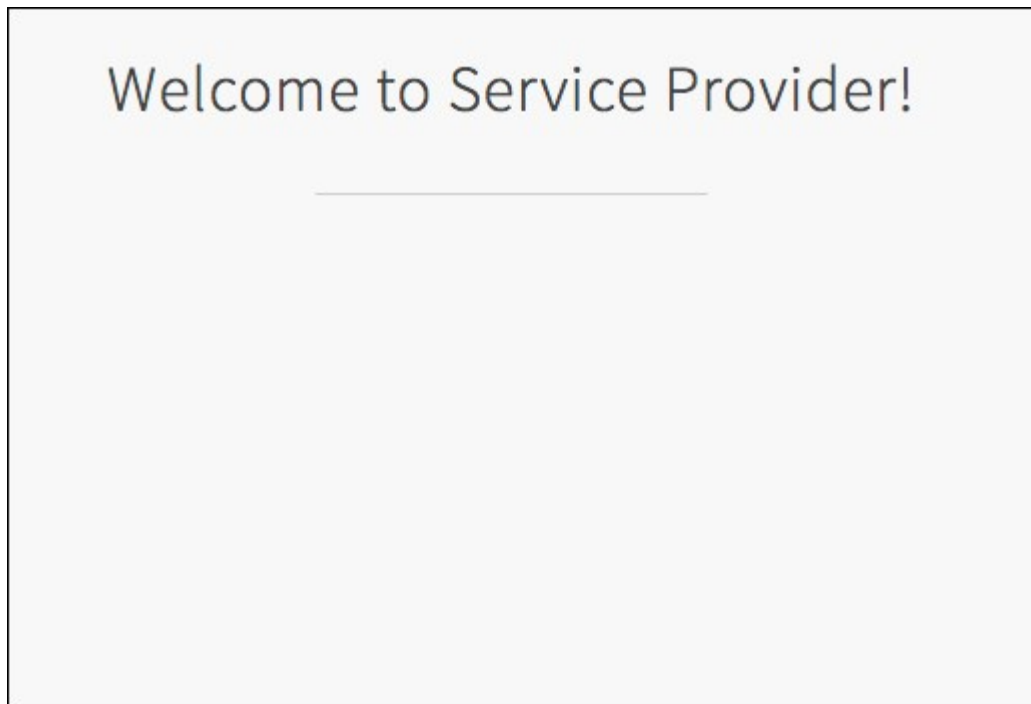
Troubleshooting Plan-to-Plan OIDC Integration



Warning: Single Sign-On for PCF v1.6 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic explains how to resolve common errors that can arise when you configure a Single Sign-On (SSO) partnership between two SSO service plans, one acting as an Identity Provider (IDP) and one acting as a Relying Party (RP).

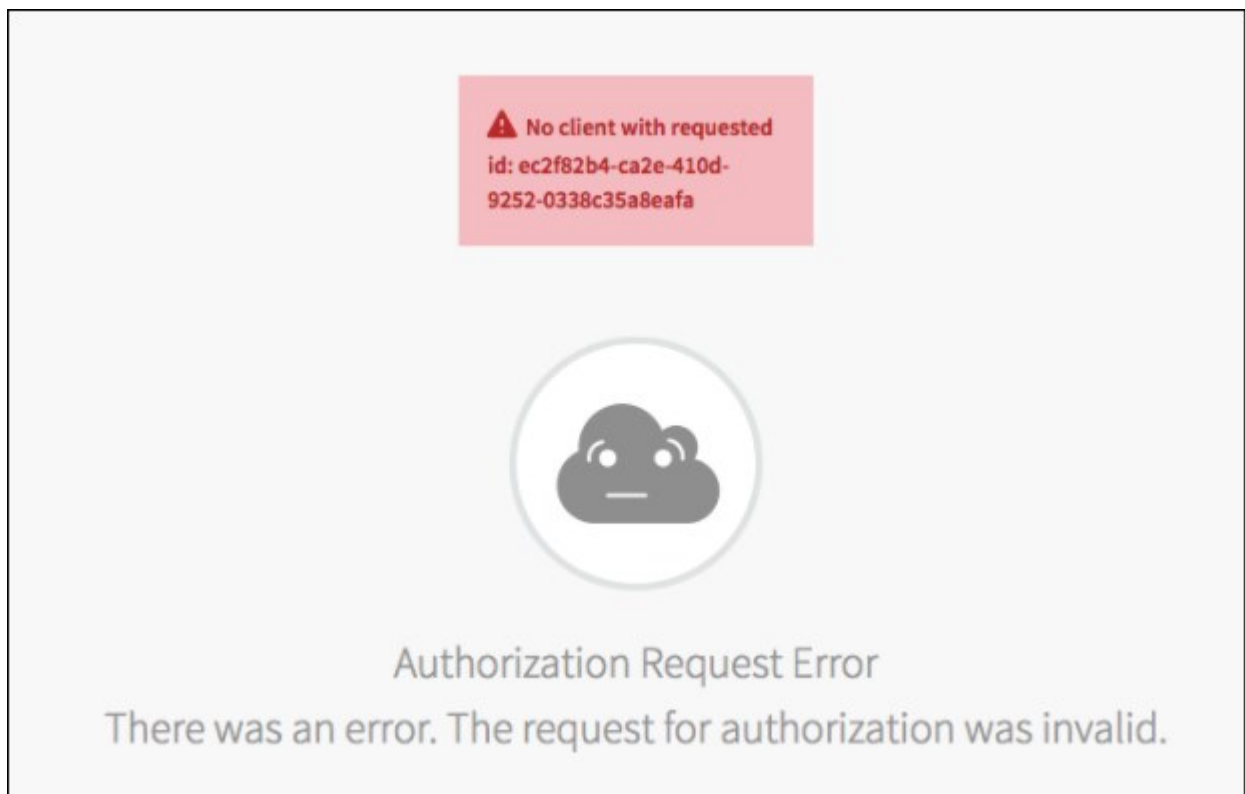
No link for OIDC, or the Service Provider Login page is blank



Cause

- The discovery URL is incorrect or unavailable. No link appears on the login page.
- This error can occur if you do not enable **Skip SSL Connection** and the IDP service plan is on a PCF instance that uses a self-signed certificate.

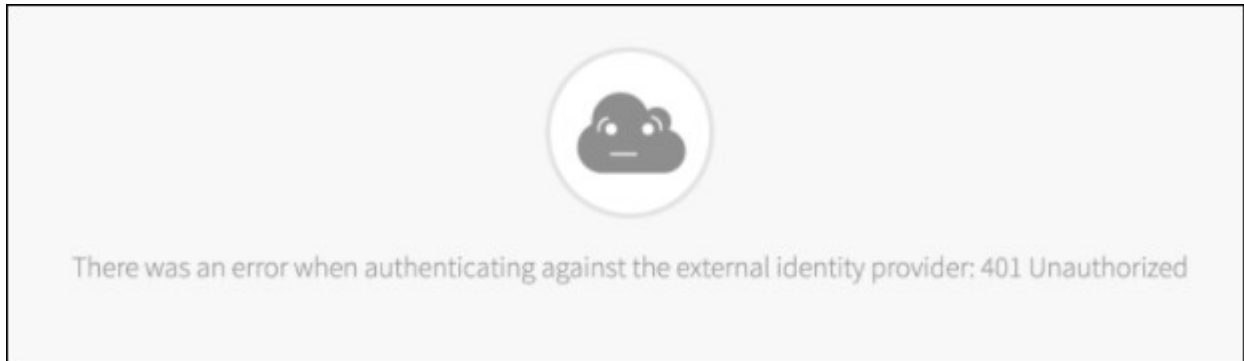
Authorization Request Error



Cause

You may have configured your OAuth client ID incorrectly.

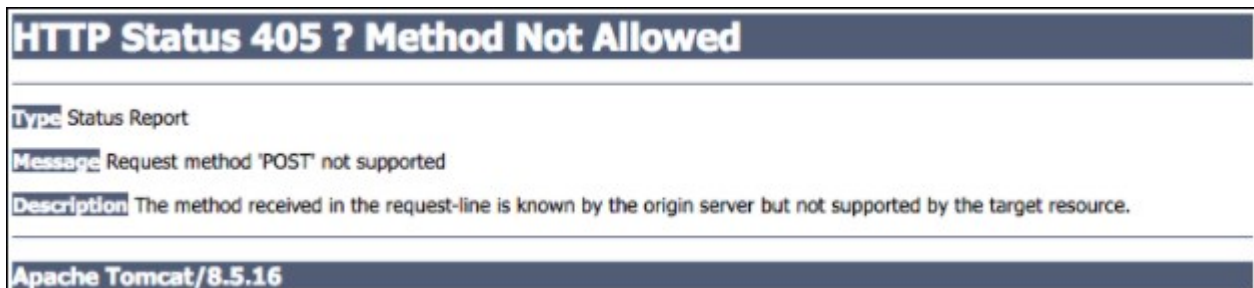
401 Unauthorized



Cause

You may have configured your OAuth client secret incorrectly.

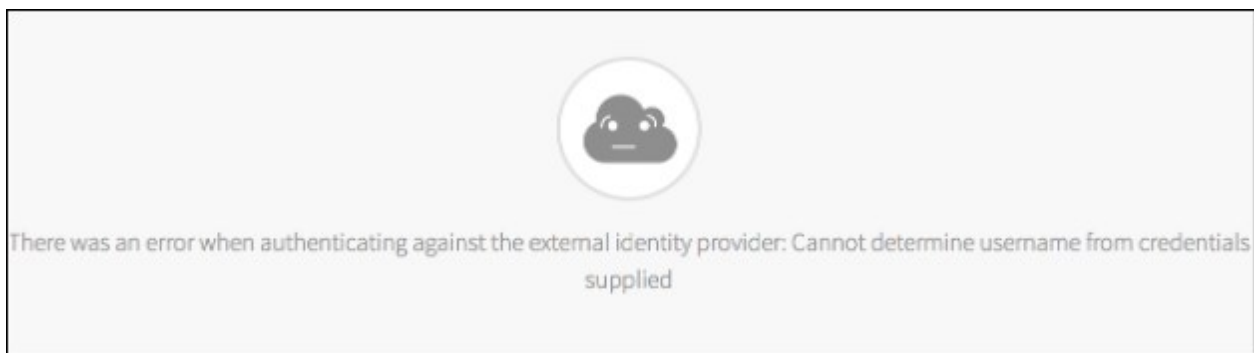
405 Method Not Allowed



Cause

- You may have omitted the `openid` scope in the IDP configuration on the RP service plan.
- You may be requesting the wrong scopes, or scopes that are not supported by the other SSO plan. Confirm that you are only requesting `openid` scopes.

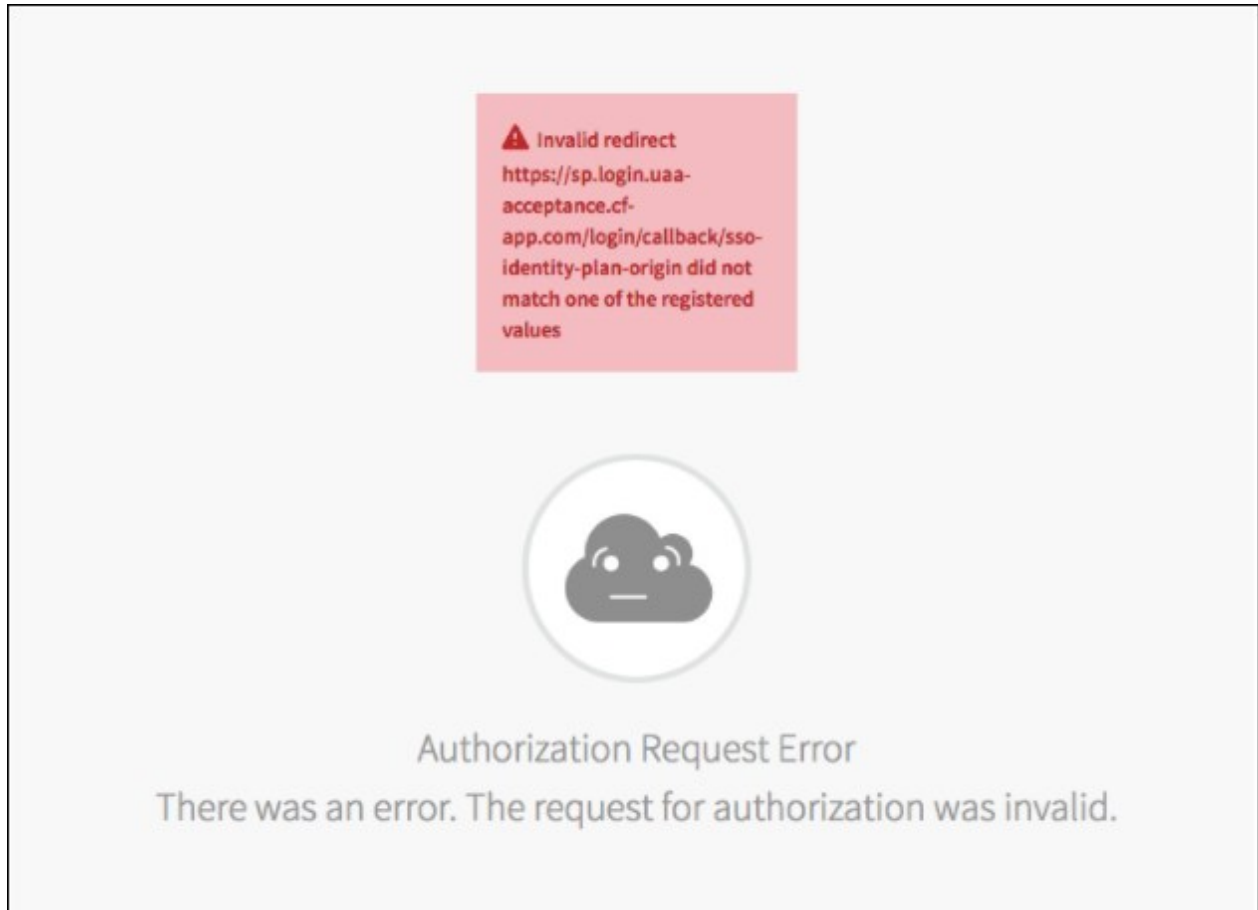
Cannot determine username with given credentials



Cause

The username you used may not have a value mapped to it. In the IDP attributes, map the “username” attribute to “username.”

Invalid redirect



Cause

You may have configured the authorized redirect URI incorrectly. Confirm that your callback URL is entered correctly as an authorized redirect URI for the client configurations on the IDP service plan.

[Create a pull request or raise an issue on the source for this page in GitHub](#)