

Single Sign-On for VMware Tanzu Application Service 1.11

Single Sign-On for VMware Tanzu Application service 1.11

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

Pivotal Single Sign-On	16
About Single Sign- On	16
OAuth 2.0 Authorization	16
Product Snapshot	17
Integration Guides	17
Useful Links	17
Release Notes	19
v1.11.3	19
Security Fixes	19
Known Issues	19
v1.11.2	19
Resolved Issue	19
Known Issues	20
v1.11.1	20
Resolved Issue	20
Known Issues	20
v1.11.0	20
Features	20
Known Issues	21
Viewing Release Notes for Another Version	21
Getting Started with Pivotal Single Sign-On	22
Install and Set Up Single Sign- On for Apps	22
Single Sign- On User Roles	22
Using Single Sign- On Components	23
Installing Pivotal Single Sign-On	24
Prerequisites	24
Install Single Sign- On Using Ops Manager	24
Update Stemcell	25
Update SSL and Load Balancer	25
Configure Application Security Groups	25
Using the System Plan	27

System Plan Best Practice	27
Administrators: Configure the System Plan for an Org	27
Developers: Create a System Plan Instance for Your App	28
Developers: Revoke System Plan Access for an Externally Hosted App	29
Managing Service Plans	30
Create or Edit Service Plans	30
Delete Service Plans	32
Automate Service Plan Creation Using Single Sign- On API	32
Create a New Pivotal Platform Admin for Single Sign- On	35
Managing Service Plan Configurations	37
Configure a Token Policy	37
Create a UAA Identity Zone Admin Client	37
Create an Admin Client	38
Create a UAA Identity Zone Admin Client	38
Update UAA Identity Zone Configurations with the API	39
Set Default Identity Provider (IDP)	41
Disable Single Sign- On Plans	42
(Optional) Modify Branding	42
(Optional) Add Default Groups for Users	43
Rotate JSON Web Token (JWT) Signing Keys	43
Configuring Internal User Store	45
Overview	45
Configure the Internal User Store	45
Add Internal Users Using UAAC	47
Test Identity Provider Configurations	48
Configuring External Identity Providers	49
Configure an External Identity Provider	49
Add a SAML Provider	49
Configure SAML Settings	51
Add an OIDC Provider	52
Add an LDAP Provider	54
Delete an External Identity Provider	57
Transfer Group Membership and Roles	57
Create or Edit Resource Permissions Mapping	57
Delete Resource Permissions Mapping	58
Configure Group Allowlist for an External Identity Provider	59

Test Identity Provider Configurations	59
Updating Identity Providers with UAAC	60
Create a UAA Identity Zone Admin Client	60
Create an Admin Client	60
Create a UAA Identity Zone Admin Client	60
Update UAA Identity Provider Configurations with the API	62
Enable Client Auth for OpenID Connect (OIDC)	64
Enable Password Grant for OpenID Connect (OIDC)	65
Skip SSL Validation for SAML	65
Managing Clients with UAAC	66
About Managing Clients with UAAC	66
When Not to Use UAAC	66
When to Use UAAC	66
Create an Admin Client	66
Enabling Identity Provider Discovery	68
What IdP Discovery Does	68
Example	68
Enable IdP Discovery	68
Managing Users	72
Manage Users in an Internal User Store	72
Manage Users from an External Identity Provider	74
Manage Users with the UAA CLI	75
About Performance	78
Monitoring Service Plans and Apps	79
Overview	79
Monitor Single Sign- On Plan Events	79
Prerequisites	79
Use the Single Sign- On API	80
Use the SSO Operator Dashboard	80
Monitor App Events	80
Backing Up and Restoring	82
Determining Pivotal Single Sign- On App Type	83
Determine Your Single Sign- On App Type	83

OAuth 2.0 Grant Types	85
Authorization Code Grant Type	85
Authorization Code Grant Type Roles	85
Authorization Code Grant Type Flow	85
Client Credentials Grant Type	87
Client Credentials Grant Type Roles	87
Client Credentials Flow	87
Resource Owner Password Grant Type	88
Resource Owner Password Grant Type Roles	88
Resource Owner Password Flow	88
Implicit Grant Type	89
Implicit Grant Type Roles	89
Implicit Flow	89
Common App Architecture Patterns	91
Externally Hosted Apps Call Pivotal Platform APIs	91
UAA Authorization Code Grant — Browser	92
SAML Bearer Token Exchange — Back End	92
Example SAML Assertion	93
JWT Bearer Token Exchange	95
Example JWT Token Content	95
Handling the JWT Token Exchange Using a Gateway	96
Set Up for SAML or JWT Bearer Token Exchange	96
Managing Service Instances	98
Create Service Instances	98
Access the SSO Developer Dashboard	98
Delete Service Instances	98
Configuring Apps	100
Overview	100
Register an App	100
Prerequisites for Registering Apps	100
Register a PAS App	101
Configure Single Sign- On Properties with JSON Bind Parameters	101
Bind Parameters	102
Configure Single Sign- On Properties with the App Manifest	103
Register an Externally Hosted App	106
Register an Externally Hosted App Using Service Keys	106
Register an Externally Hosted App Using the SSO Developer Dashboard	108

Manage App Configurations	110
Prerequisite	111
Procedure	111
Credentials of Existing App Configurations	113
Regenerate an App Secret	113
Create an Admin Client	114
Unregister App from Single Sign- On	115
Unregister a PAS App	115
Unregister an Externally Hosted App Using Service Keys	116
Unregister an Externally Hosted App Using SSO Developer Dashboard	116
Managing Resources	117
Create or Edit Resources	117
Delete Resources	118
About Space Protection for Resources	118
Integrating Pivotal Single Sign- On with Your App	120
Integrate Single Sign- On with an App	120
Java Apps	120
Non-Java Apps	120
Login Hints	121
Active Directory Federation Services Integration Guide Overview	122
Prerequisites	122
Active Directory Federation Services Integration Guide	122
Configuring AD FS with Single Sign- On	122
Testing and Troubleshooting	122
Configuring Active Directory Federation Services as an Identity Provider	124
Set Up SAML with the SSO Operator Dashboard	124
Set Up SAML in AD FS	125
Setting Up Groups in SAML from AD FS	135
Create Custom Value “groups”	137
Configuring a Single Sign-On Service Provider	141
Overview	141
Download Identity Provider Metadata	141
Create a New SAML Identity Provider	141
Configure Your New Identity Provider	142
Testing	144

Test Your Service Provider Connection	144
Test Your Identity Provider Connection	149
Test Your Single Sign-Off	150
Troubleshooting	152
Event Viewer	152
Azure Active Directory SAML Integration Guide Overview	153
Prerequisites	153
Azure AD Integration Guide	153
Configuring Azure AD with Single Sign- On	153
Testing and Troubleshooting	153
Configuring Azure Active Directory as a SAML Identity Provider	155
Step 1: Set up SAML in Single Sign- On	155
Step 2: Set up SAML in Azure AD	156
Step 3: Set up Claims Mapping	159
Configuring a Single Sign-On Service Provider	161
Step 1: Set up SAML	161
Step 2: Configure Group Permissions	162
Testing	164
Test Your Configurations in Azure AD	164
Test Your Service Provider Connection	165
Test Your Identity Provider Connection	170
Test Your Single Sign-Off	171
Troubleshooting	173
Failed Login	173
Symptom:	173
Solutions:	173
App ID Not Found	173
Symptom:	173
Explanations:	174
Reply URL Does Not Match	174
Symptom:	174
Explanation:	174
Missing Name ID	174
Symptom:	175
Explanation:	175

Azure Active Directory OIDC Integration Guide Overview	176
Prerequisites	176
Azure AD Integration Guide	176
Configuring Azure AD with Single Sign- On	176
Testing and Troubleshooting	176
Configuring Azure Active Directory as an OIDC Identity Provider	178
Overview	178
Prerequisites	178
Set Up a Relying Party in Azure AD	178
Register a New App	178
Generate a Relying Party OAuth Client Secret	180
Configure Reply and Endpoint URLs	181
Set Up the OIDC Identity Provider in Single Sign- On	184
Testing Your Single Sign-On Connection	186
Troubleshooting	190
Bad Request	190
Symptom:	190
Cannot determine username from credentials supplied	190
Symptom:	190
Explanation:	191
Azure Error for Reply Address	191
Symptom:	191
Explanation:	192
Login Page Cannot Be Found (404 Error)	192
Symptom:	193
Explanation:	193
Error authenticating against external identity provider: 404 Not Found	193
Symptom:	193
Explanation:	193
Error authenticating against external identity provider: Invalid issuer for token did not match expected	193
Symptom:	193
Explanation:	194
Request Method 'POST' not supported (405 Error)	194
Symptom:	194
Explanation:	194

Error authenticating against external identity provider: Some parties were not in the token audience	194
Symptom:	194
Explanation:	194
Layer7 SiteMinder Integration Guide Overview	195
Prerequisites	195
Layer7 SiteMinder Integration Guide	195
Configuring Layer7 SiteMinder with Single Sign- On	195
Testing and Troubleshooting	195
Configuring Layer7 SiteMinder as an Identity Provider	197
Set up SAML in Pivotal Single Sign- On	197
Set up SAML in Layer7 SiteMinder	198
Configuring a Single Sign-On Service Provider	203
Set up SAML	203
Testing	205
Test Your Service Provider Connection	205
Test Your Identity Provider Connection	209
Test Your Single Sign-Off	210
Troubleshooting	212
Layer7 SiteMinder Partnership is Inactive	212
Symptom:	212
Explanations:	212
Service Provider Entity ID Misconfigured	212
Symptom:	212
Explanation:	212
Incoming SAML message is invalid	212
Symptom:	212
Explanation:	213
Assertion Consumer Service URL Misconfigured	213
Symptom:	213
Explanation:	213
Audience Field Misconfigured	213
Symptom:	213
Explanation:	213
Expired Certificate	213
Symptom:	213

Explanation:	213
Identity Provider SSO URL Misconfigured	214
Symptom:	214
Explanation:	214
Google Cloud Platform OIDC Integration Guide Overview	215
Prerequisites	215
Integrate Google Cloud Platform OIDC for Single Sign- On	215
Test and Troubleshoot	215
Configuring GCP as an OIDC Identity Provider	216
Overview	216
Generate GCP Client Credentials	216
Set up the OIDC Identity Provider in Single Sign- On	218
Testing	220
Test Your Single Sign-On Connection	220
Troubleshooting	223
No Link for OIDC	223
Symptom:	223
Explanation:	223
No OAuth Client Found	223
Symptom:	223
Explanation:	224
Unauthorized	224
Symptom:	224
Explanation:	224
Redirect URI Mismatch	224
Symptom:	224
Explanation:	225
Empty Username	225
Symptom:	225
Explanation:	225
Unable to map claim to a username	225
Symptom:	225
Explanation:	226
Okta Integration Guide Overview	227
Prerequisites	227
Okta Integration Guide	227

Configuring Okta with Single Sign- On	227
Testing and Troubleshooting	227
Configuring Okta as an Identity Provider	229
Set up SAML in Single Sign- On	229
Set Up SAML in Okta	230
Configuring a Single Sign-On Service Provider	233
Setting up SAML	233
Testing	236
Test Your Service Provider Connection	236
Test Your Identity Provider Connection	240
Test Your Single Sign-Off	241
Troubleshooting	243
Page Not Found	243
Symptom:	243
Explanations:	243
No Valid Assertion	243
Symptom:	243
Explanations:	244
Webpage Not Available	244
Symptom:	244
Explanation:	244
Metadata Not Found	244
Symptom:	244
Explanation:	245
PingFederate Integration Guide Overview	246
Prerequisites	246
PingFederate Integration Guide	246
Configuring PingFederate with Single Sign- On	246
Testing and Troubleshooting	246
Configuring PingFederate as an Identity Provider	248
Set up SAML in Single Sign- On	248
Set up SAML in PingFederate	249
Configure the Connection	249
Configure Browser SSO	250
Assertion Creation	251

Protocol Settings	252
Configure Credentials	253
Configuring a Single Sign-On Service Provider	255
Set up SAML	255
Testing	257
Test Your Service Provider Connection	257
Test Your Identity Provider Connection	261
Test Your Single Sign-Off	262
Troubleshooting	264
Error	264
Symptom:	264
Explanations:	264
Metadata Not Found	264
Symptom:	265
Explanation:	265
PingOne Cloud Integration Guide Overview	266
Prerequisites	266
PingOne Cloud Integration Guide	266
Configuring PingOne Cloud with Single Sign- On	266
Testing and Troubleshooting	266
Configuring PingOne Cloud as an Identity Provider	268
Set up SAML in Single Sign- On	268
Set up SAML in PingOne Cloud	269
Configuring a Single Sign-On Service Provider	272
Set up SAML	272
Testing	274
Test Your Service Provider Connection	274
Test Your Identity Provider Connection	279
Test Your Single Sign-Off	280
Troubleshooting	282
Error	282
Symptom:	282
Explanations:	282

Something went amiss	282
Symptom:	282
Explanation:	283
Metadata Not Found	283
Symptom:	283
Explanation:	283
Missing Name ID	283
Symptom:	283
Explanation:	284
 Plan-to-Plan OIDC Integration Guide	 285
Prerequisites	285
Integrating a Plan-to-Plan OIDC for Single Sign- On	285
Testing the OIDC Connection	285
Troubleshooting	286
 Configuring Plan-to-Plan OIDC Integration	 287
Overview	287
Prerequisites	287
Set Up Relying Party Configurations in the Identity Provider Plan	287
Set Up the OIDC Identity Provider Configuration in the Relying Party Plan	289
Finish Configuration	290
 Testing OIDC Integrations	 291
Testing Your Single Sign- On Connection	291
 Troubleshooting Plan-to-Plan OIDC Integration	 293
No link for OIDC, or the Service Provider Login page is blank	293
Cause	293
Authorization Request Error	293
Cause	294
401 Unauthorized	294
Cause	294
405 Method Not Allowed	294
Cause	295
Cannot determine username with given credentials	295
Cause	295
Invalid redirect	295
Cause	296

Pivotal Single Sign-On



Warning: Pivotal Single Sign-On v1.11 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.



Note: Pivotal has renamed *Single Sign- On for Pivotal Cloud Foundry* to *Pivotal Single Sign- On*.



Note: Pivotal has renamed *Pivotal Cloud Foundry* to *Pivotal Platform*.

This topic provides an overview of Pivotal Single Sign- On.

Single Sign- On is an all-in-one solution for securing access to apps and APIs on Pivotal Platform. Single Sign- On 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, Single Sign- On uses OAuth 2.0 to secure resources or APIs.

About Single Sign- On

Single Sign- On enables users to log in through a single sign-on service and access other apps that are hosted or protected by the service. This improves security and productivity by removing the need for users to log in to individual apps.

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 apps and APIs.

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

OAuth 2.0 Authorization

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

Developers define resources required by an application bound to a Single Sign- On 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:

Element	Details
Version	1.11.3
Release date	October 28, 2020
Compatible Ops Manager version(s)	2.6, 2.7, and 2.8
Compatible Pivotal Application Service version(s)	2.6, 2.7, and 2.8
IaaS support	AWS, GCP, OpenStack, Azure, and vSphere

Integration Guides

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

- [Active Directory Federation Services \(AD FS\) Integration Guide](#)
- [Azure Active Directory SAML Integration Guide](#)
- [Azure Active Directory OIDC Integration Guide](#)
- [Layer7 SiteMinder 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: Pivotal Single Sign-On v1.11 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.



Note: Pivotal has renamed *Single Sign- On for Pivotal Cloud Foundry* to *Pivotal Single Sign- On*.



Note: Pivotal has renamed *Pivotal Cloud Foundry* to *Pivotal Platform*.

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

For product versions and upgrade paths, see [Upgrade Planner](#).

v1.11.3

Release Date: October 28, 2020

Security Fixes

This release includes the following security fix:

- High [CVE-2020-5425: User Impersonation possible in Tanzu SSO](#)

Known Issues

This release has the following issue:

- **Authorization for Okta OpenID Connect (OIDC):** When using an Okta OIDC provider, the roles claim in the ID token does not get populated with external identity provider (IdP) groups. This impacts the mapping of external IdP groups to scopes. Despite this limitation, you can still use Okta OIDC provider for authentication.
- Users are unexpectedly logged out from SSO Operator Dashboard.

v1.11.2

Release Date: April 24, 2020

Resolved Issue

This release has the following fix:

- Space Developers can now view and select permissions from any space in the SSO Developer Dashboard on the **Register App** page.

Known Issues

This release has the following issue:

- **Authorization for Okta OpenID Connect (OIDC):** When using an Okta OIDC provider, the roles claim in the ID token does not get populated with external identity provider (IdP) groups. This impacts the mapping of external IdP groups to scopes. Despite this limitation, you can still use Okta OIDC provider for authentication.
- Users are unexpectedly logged out from SSO Operator Dashboard.

v1.11.1

Release Date: April 13, 2020

Resolved Issue

This release has the following fix:

- Space Developers can now select the `profile` scope in the SSO Developer Dashboard on the **Register App** page.

Known Issues

This release has the following issue:

- **Authorization for Okta OpenID Connect (OIDC):** When using an Okta OIDC provider, the roles claim in the ID token does not get populated with external identity provider (IdP) groups. This impacts the mapping of external IdP groups to scopes. Despite this limitation, you can still use Okta OIDC provider for authentication.
- Users are unexpectedly logged out from SSO Operator Dashboard.
- Space Developers cannot view and select permissions from other spaces in the SSO Developer Dashboard on the **Register App** page.

v1.11.0

Release Date: December 11, 2019

Features

New features and changes in this release:

- **Authentication Field:** Developers can use the SSO Developer Dashboard to set the permissions that users must have to log in to an app. For more information, see [Register an Externally Hosted App Using the SSO Developer Dashboard](#).
- **Space Auditor Role:** Developers with the Space Auditor role can view the SSO Developer Dashboard. They cannot edit any configurations. For more information, see [Pivotal Single](#)

[Sign- On User Roles.](#)

Known Issues

This release has the following issues:

- **Authorization for Okta OpenID Connect (OIDC):** When using an Okta OIDC provider, the roles claim in the ID token does not get populated with external identity provider (IdP) groups. This impacts the mapping of external IdP groups to scopes. Despite this limitation, you can still use Okta OIDC provider for authentication.
- Users are unexpectedly logged out from SSO Operator Dashboard.
- Space Developers cannot view and select permissions from other spaces in the SSO Developer Dashboard on the **Register App** page.
- Space Developers cannot select the `profile` scope in the SSO Developer Dashboard on the **Register App** page.

Viewing Release Notes for Another Version

To view the release notes for another product version, select the version from the dropdown 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 Pivotal Single Sign-On



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On.

Install and Set Up Single Sign-On for Apps

1. [Install Single Sign-On](#) using Ops Manager.
2. [Create a Service Plan](#). Single Sign-On is a multi-tenant service and a service plan corresponds to a tenant. This enables 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 Platform 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.

Single Sign-On User Roles

User roles determine the parts of a Single Sign-On configuration that a user can view or manage. Single Sign-On uses the following user roles:

- **Pivotal Platform Administrators**, who can manage service plans, service instances, identity providers (IdPs), apps, and resources. This is a Pivotal Platform user role.
- **Plan Administrators**, who can manage service instances, IdPs, apps, and resources. This is a user role that is specific to Single Sign-On.
- **Space Developers**, who can manage service instances, apps, and resources. This is a Pivotal

Platform user role.

- **Space Auditors**, who can view service instances, apps, and resources. They cannot edit any configurations. This is a Pivotal Platform user role.

The following table shows the permissions for each role:

Access by role	Pivotal Platform Administrator	Plan Administrator	Space Developer	Space Auditor
Service plans	M			
Service instances	M	M	M	V
Identity providers	M	M		
Applications	M	M	M	V
Resources	M	M	M	V
Legend: M = Manage, V = View				

Using Single Sign- On Components

In addition to apps, Single Sign- On supports single sign-on for components of Pivotal Platform, including Ops Manager and Apps Manager. This enables users already managed in an external IdP to sign into Pivotal services.

Refer to the following pages for instructions on configuring Single Sign- On to enable users in an external identity store to access Pivotal Platform 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](#)

Installing Pivotal Single Sign-On



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On.

Prerequisites

To install Single Sign-On, you must have:

- [Ops Manager](#)
- [Pivotal Application Service](#)
- SSL certificates
- Application Security Groups (ASGs)

Install Single Sign-On Using 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 + 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 Identity Service Broker is deployed as an app 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 Pivotal Application Service network.

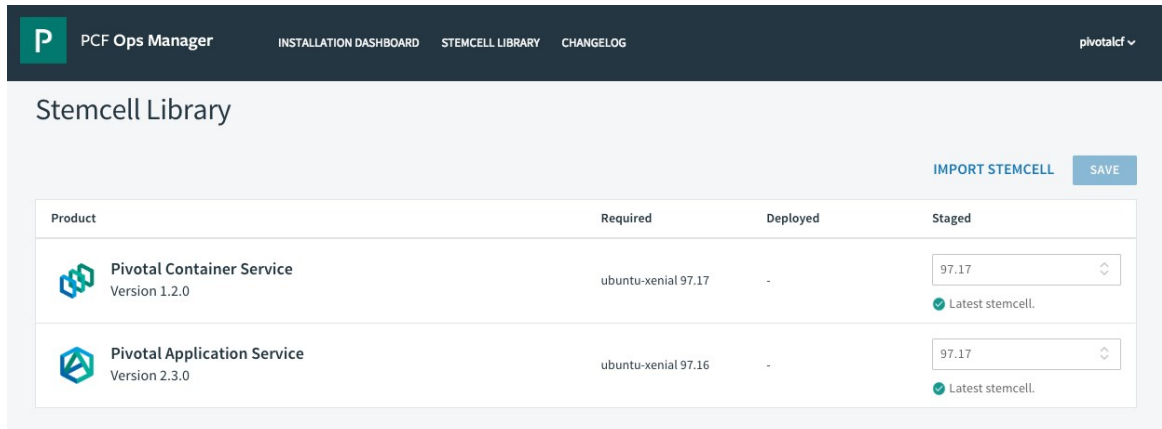
5. In the Ops Manager Dashboard, do the following to complete the installation:
 1. If you are using Ops Manager v2.3 or later, click **Review Pending Changes**. For more information about this Ops Manager page, see [Reviewing Pending Product Changes](#).

2. Click **Apply Changes**.

Update Stemcell

If required, do the following to update the stemcell for Single Sign- On:

1. Download the stemcell from [Pivotal Network](#).
2. In the Ops Manager, click **Stemcell Library**.
3. Click **Import Stemcell**, and then select the stemcell you downloaded from Pivotal Network.



4. Click **Save**.

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

Single Sign- On 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 Single Sign- On, you must ensure your ASG allows access to the load balancers 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On. The system plan is the default plan meant for developer apps, not end-user apps.

Single Sign-On 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 Pivotal Platform 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 Single Sign-On 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 Single Sign-On 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

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

1. Log into the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your User Account and Authentication (UAA) administrator credentials. In your PAS 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).

The screenshot shows the 'System' plan configuration page in the SSO Operator Dashboard. The page has a dark header with a 'P' logo, 'Single Sign-On' text, and a user dropdown 'admin'. Below the header is a breadcrumb 'Plans > System'. The main content area is titled 'System' and contains the following fields:

- Plan Name***: A text input field containing 'System'.
- Description***: A text input field containing 'This plan is reserved for app developer single sign-on.' Below this field is a note: 'This will appear as a plan feature in the Apps Manager Marketplace'.
- Auth Domain***: A text input field containing 'https://login.sys.banana.gcp.releng.cf-app.com'.
- Organizations**: A section with a search input field containing 'MY-ORG' and a dropdown menu showing 'MY-ORG'. To the right, there is a list of organizations with 'sree-org' and a close button 'x'.

At the bottom right of the form are two buttons: 'Cancel' and '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 Single Sign- On.
2. If you have a Pivotal Platform app, bind the application with the service instance you created. For more information, see [Register a Pivotal Platform App](#).
3. If your app is a pipeline or a script that runs external to Pivotal Platform but calls Pivotal Platform 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 Developer Dashboard. Copy the login portal URL into a browser, and log in using your UAA Admin Credentials.
 3. Copy the **Temporary Authentication Code** from the browser into the UAAC to finish the authentication.
 4. Run `uaac context`.
 5. 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 Externally Hosted App

To revoke system plan access from an externally hosted app that is registered with the system plan to access Pivotal Platform 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: Pivotal Single Sign-On v1.11 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 Platform admins manage Single Sign-On service plans.

Pivotal Single Sign-On is a multi-tenant service, which enables a deployment to host multiple tenants as service plans. Each service plan can have its own admins, apps, 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 Single Sign-On as an OpenID Connect (OIDC) identity provider. For more information, see [Plan-to-Plan OIDC Integration Guide](#).

Create or Edit Service Plans

Pivotal Platform admins can create new Single Sign-On service plans at any time from the SSO Operator Dashboard. You can use the SSO Operator Dashboard to create and configure service plans at any time.



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

1. Log into the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your User Account and Authentication (UAA) admin credentials. You can find these credentials in the Pivotal Application Service tile in Ops Manager in the **Credentials** tab.
2. Click **New Plan** on the SSO Operator Dashboard to create a new Single Sign-On 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 hostname where users authenticate to access apps 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.



Note: You cannot add system operators to this list. System operators do not appear in this list because they already have Plan Administrator privileges.

8. Under **Organizations**, select specific orgs in your Pivotal Platform 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 org. This is only recommended for test plans to allow developers to experiment with Single Sign- On.

- ✦ If you do not select any orgs, 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 orgs you selected. Users in those orgs 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 Single Sign- On database all of the configurations, identity providers, users, app configurations, and resources associated with the plan. It also deletes the associated service instances and service bindings. You must rebind any apps bound to the deleted service instances to new service instances.

1. Log in to the SSO Operator Dashboard at `https://p-identity.SYSTEM-DOMAIN` using your UAA admin credentials. You can find these credentials in your 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 dropdown.
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.

Automate Service Plan Creation Using Single Sign- On API

Pivotal Platform admins can create new Single Sign- On service plans using the Single Sign- On API. This allows them to automate creating and deleting Single Sign- On plans. Pivotal recommends creating a dedicated client for Single Sign- On plan automation.

To automate service plan creation:

1. To install the UAA CLI, run:

```
gem install cf-uaac
```

2. To target your Pivotal Platform UAA server, run:

```
uaac target uaa.SYSTEM-DOMAIN
```

3. To record your admin credentials, do one of the following:

- ✦ Obtain **Admin Client Credentials** from Ops Manager.
- ✦ Obtain **uaa:admin:client_secret** from your deployment manifest.

4. To authenticate and obtain an access token for the admin client from the UAA server, run:

```
uaac token client get admin -s ADMIN-CLIENT-SECRET
```

Where `ADMIN-CLIENT-SECRET` is the admin credentials you recorded in step 3.

UAAC stores the token in `~/.uaac.yml`.

- To create an automation client with UAAC, run:

```
uaac client add AUTO-CLIENT-ID --secret AUTO-CLIENT-SECRET \
--authorized_grant_type client_credentials \
--authorities "cloud_controller.admin,zones.write,scim.write,scim.read"
```

Where:

- ♦ `AUTO-CLIENT-ID` is the name of the automation client you want to use.
- ♦ `AUTO-CLIENT-SECRET` is the secret for the automation client you want to use.

- To obtain an access token for your automation client, run:

```
uaac token client get AUTO-CLIENT-ID -s AUTO-CLIENT-SECRET
```

Where:

- ♦ `AUTO-CLIENT-ID` is the name you provided in step 5.
- ♦ `AUTO-CLIENT-SECRET` is the secret you provided in step 5.

- To obtain your automation access token, run:

```
uaac context
```

For example:

```
$ uaac context

[1]*[my-auto-client]
client\_id: my-client-id
access\_token: aBcdEfg0hIJKlm123.e
token\_type: bearer
expires\_in: 43200
scope: cloud\_controller.admin zones.write scim.write scim.read
jti: 91b3-abcd1233
```

- Record the `access_token` value from the output of the previous step.
- To create a new Single Sign-On plan and record the plan ID, run:

```
curl -X POST "https://sso-api.SYSTEM-DOMAIN/v1/plans" \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "name": "PLAN-NAME",
  "description": "DESCRIPTION",
  "auth_domain": "AUTH-DOMAIN",
  "instance_name": "INSTANCE-NAME"
}'
```

Where:

- ✦ **TOKEN** is the access token you recorded from the previous step.
- ✦ **PLAN-NAME** is the name of your plan.
- ✦ **DESCRIPTION** is the text you want to appear as a plan feature in the Services Marketplace.
- ✦ **AUTH-DOMAIN** is the **Auth Domain** you entered in [Create or Edit Service Plans](#).
- ✦ **INSTANCE-NAME** is the name of your instance. This text appears in user-facing content, such as email communications.

The above command returns output similar to the following:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "1",
  "name": "some-plan-name",
  "description": "some-description",
  "auth\_domain": "some-auth-domain",
  "instance\_name": "some-instance-name"
}
```

- Record the **id** value from the output of the previous step.

Alternatively, you can save the plan ID, by parsing the output from the previous step. For example, you can run:

```
$PLAN-ID=$(curl -X POST "https://sso-api.SYSTEM-DOMAIN/v1/plans" \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "name": "PLAN-NAME",
  "description": "DESCRIPTION",
  "auth_domain": "AUTH-DOMAIN",
  "instance_name": "INSTANCE-NAME"
}' | jq -r '.id')
```



Note: Using **curl** instead of **uaac curl** in the example to facilitates parsing the response for ID below.

- To add plan administrators, who can view the plan and manage identity providers, run the following command for each plan administrator:

```
uaac member add zones.PLAN-ID.admin USER-NAME
```

Where:

- ✦ **PLAN-ID** is the **id** recorded in the previous step.
- ✦ **USER-NAME** is the username of the plan administrator you are adding.

- To authenticate as your automation client, run:

```
cf api api.SYSTEM-DOMAIN
cf auth AUTO-CLIENT-ID AUTO-CLIENT-SECRET --client-credentials
```

Where:

- ✦ `AUTO-CLIENT-ID` is the name you provided in step 5.
- ✦ `AUTO-CLIENT-SECRET` is the secret you provided in step 5.

13. To give orgs access to your Single Sign- On plan, do one of the following:

- ✦ To give specific orgs access to your Single Sign- On plan, run the following command for each org:

```
cf enable-service-access p-identity -p AUTH-DOMAIN -o ORG-NAME
```

Where `ORG-NAME` is the name of the org you want to have access to your Single Sign- On plan.

- ✦ To give all orgs access to your Single Sign- On plan, run:

```
cf enable-service-access p-identity -p AUTH-DOMAIN
```



Pivotal recommends only giving all orgs access to your Single Sign- On plans for test plans to enable developers to experiment with Single Sign- On.

For more information on how you can manage Single Sign- On plans using the Single Sign- On API, see the [Single Sign- On API](#) documentation.

Create a New Pivotal Platform Admin for Single Sign- On

Pivotal Platform admins can grant users additional permissions to allow them to manage Single Sign- On plans. These permissions let users act as Pivotal Platform admins.



Warning: If you use external group mappings, create group mappings for these scopes instead. If you follow the below procedure, permissions are directly assigned to your users. For more information, see [Grant Admin Permissions to an External Group \(SAML or LDAP\)](#).

To create a new Pivotal Platform admin:

1. To install the UAA CLI, run:

```
gem install cf-uaac
```

2. To target your Pivotal Platform UAA server, run:

```
uaac target uaa.SYSTEM-DOMAIN
```

3. To record your admin credentials, do one of the following:

- ❖ Obtain **Admin Client Credentials** from OpsManager.
 - ❖ Obtain **uaa:admin:client_secret** from your deployment manifest. UAAC stores the token in `~/.uaac.yml`.
4. To authenticate and obtain an access token for the admin client from the UAA server, run:

```
uaac token client get admin -s ADMIN-CLIENT-SECRET
```

Where `ADMIN-CLIENT-SECRET` is the admin credentials you recorded in step 3.

5. To allow users to manage Single Sign-On plans, run:

```
uaac member add cloud_controller.admin ADMIN-USERNAME
uaac member add scim.read ADMIN-USERNAME
uaac member add zones.read ADMIN-USERNAME
uaac member add zones.write ADMIN-USERNAME
```

Where `ADMIN-USERNAME` is the username of the user you want to make an admin.

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

Managing Service Plan Configurations



Warning: Pivotal Single Sign-On v1.11 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 Platform admins or plan admins can manage Pivotal Single Sign-On service plan configurations using the SSO Operator Dashboard or API using the User Account and Authentication Command Line Interface (UAAC).

Single Sign-On manages configurations within the UAA and the Cloud Controller (CC) components of the Pivotal Application Service. Each Single Sign-On service plan ties together a CC plan and a UAA identity zone.

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

Configure a Token Policy

Single Sign-On enables Pivotal Platform admins and plan admins 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:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your tile in Ops Manager in **Credentials**.
2. Select the name of the plan that you want to configure a token policy for, and click **Configure** from the dropdown.
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 UAA Identity Zone Admin Client

To use the UAAC with your Pivotal Single Sign-On 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 Single Sign-On service plan.

Create an Admin Client

To create a UAA admin client:

1. Follow the procedure in [Create an Admin Client](#).
2. 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 Operator Dashboard 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: asdioqwuek12312.e21e
  token_type: bearer
  expires_in: 43200
  scope: uaa.admin
  jti: 123908dk11-23298
```

You can now do operator level API configurations for the Single Sign- On service plan. You do not have permissions for any other Single Sign- On 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 [Cloud Foundry documentation](#).

To make UAA identity zone API calls:

1. Create an identity zone admin client by following the procedure in [Create a UAA Identity Zone Admin Client](#) above.
2. Find the UAA identity zone ID:
 - a. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your User Account and Authentication (UAA) admin credentials. You can find these credentials in the tile in Ops Manager in **Credentials**.
 - b. Click the name of the Single Sign-On service plan that you want to configure on the SSO Operator Dashboard, and select **Edit Plan** from the dropdown.
 - c. Record the identity zone ID for your plan from the SSO Operator Dashboard URL. The URL looks similar to the below:

```
https://p-identity.SYSTEM-DOMAIN/dashboard/edit_plan/YOUR-IDENTITY-ZONE-ID
```

Where [YOUR-IDENTITY-ZONE-ID](#) is your plan's identity zone ID.

3. Direct the output to a text file by running:

```
uaac curl -k /identity-zones/YOUR-IDENTITY-ZONE-ID > JSON-BLOB.txt
```

Where:

- ✦ [YOUR-IDENTITY-ZONE-ID](#) is the UAA identity zone ID you obtained in step 2c.
 - ✦ [JSON-BLOB.txt](#) is the name of your text file.
4. 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 [YOUR-IDENTITY-ZONE-ID](#).

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

```
{
  "id": "d324e405-4976-49a4-a142-cf33e19d4715",
  "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
}
```

5. In your `JSON-BLOB.txt` file, 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.

6. 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/YOUR-IDENTITY-ZONE-ID -X PUT
-H 'Content-Type: application/json' -d "$(cat file.txt)"
```

Where `YOUR-IDENTITY-ZONE-ID` is the UAA identity zone ID you obtained in step 2c.

A truncated example command looks similar to the below:

```
$ uaac curl -k identity-zones/YOUR-IDENTITY-ZONE-ID \
-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](#).

Set Default Identity Provider (IDP)

For Pivotal Platform v2.4 or later, a default IDP can be set so that end users are automatically redirected to an appropriate enterprise IDP.

To set a default IDP:

1. Follow steps 1 – 6 in [Update UAA Identity Provider Configurations with the API](#) above, and in step 5, add the following line to the config section in the JSON blob:

```
"defaultIdentityProvider": "YOUR-IDP"
```

Where **YOUR-IDP** is the IDP you want to set as the default.

For information about `defaultIdentityProvider`, see [Creating an identity zone](#) in the UAA documentation.

Disable Single Sign- On Plans

For Pivotal Platform v2.4 and later, Single Sign- On plans that are no longer in use can be disabled. Disabled plans can be re-enabled later when they need to be used again.

To disable Single Sign- On plans:

1. Follow steps 1 – 6 in [Update UAA Identity Provider Configurations with the API](#) above, and in step 5, add the following line to the config section in the JSON blob:

```
"active": false
```

For information about `active`, see [Updating an Identity Zone](#) in the UAA documentation.

(Optional) Modify Branding

Optionally, you can modify the branding on your login page such as your company name, logos, legal text, and legal links.

To modify branding of the login page:

1. Follow steps 1 – 6 in [Update UAA Identity Provider Configurations with the API](#) above, and in step 5, add or modify the branding section in the JSON blob according to the [Cloud Foundry 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": "VMware",
  "productLogo": "(base64 of png image here, will show up as image on login p
age)",
  "squareLogo": "(base64 of png image here, will show up as browser icon)",
  "footerLegalText": "©2017 VMware, Inc. or its affiliates All Rights Reserve
d.",
  "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"
  }
},
```

(Optional) Add Default Groups for Users

You can add additional default groups for all users. You do not need to manually assign groups 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.

To add default groups for users:

1. Follow the steps 1 – 6 in [Update UAA Identity Provider Configurations with the API](#) above, and in step 5, update the default groups section in the JSON blob according to the [Cloud Foundry documentation](#). An example default groups section is shown below:



Note: You can add more groups to the array list. Users 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



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

To rotate JWT signing keys:

1. Generate a private key that can be used for signing. This is typically an asymmetric PEM-encoded private key that begins with `-----BEGIN RSA PRIVATE KEY-----`. The UAA might support other key types. For more information, see the [Cloud Foundry documentation](#).

Two example commands are below:

```
$ ssh-keygen -t rsa -m PEM -b 2048 -f OUTPUT-FILE-NAME
```

After running the above command, enter a blank password when prompted.

```
$ openssl genrsa -out OUTPUT-FILE-NAME 2048
```

Generate your signing keys securely. Ask your security organization for acceptable key generation practices.



Warning: Do not use private keys that begin with `-----BEGIN OPENSSH PRIVATE KEY-----`

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/yWp3lCElSqnKPJbCGulDQThB47kGQjBoXL8TcrkxuCyuxaV7B5ryq3w+g3R1x
-----END RSA PRIVATE KEY-----
```

Becomes:

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

3. Follow the steps 1 – 6 in [Update UAA Identity Provider Configurations with the API](#) above, and in step 5, update the token policy section in `JSON-BLOB.txt` to add your new, generated private key as the value for `signingKey`. An example of this section is shown below:

```
{
  "config": {
    "tokenPolicy": {
      "accessTokenValidity": -1,
      "refreshTokenValidity": -1,
      "jwtRevocable": false,
      "refreshTokenUnique": false,
      "refreshTokenFormat": "jwt",
      "activeKeyId": "first-signing-key",
      "keys" : {
        "first-signing-key" : {
          "signingKey" : "-----BEGIN RSA PRIVATE KEY-----\nMIIEogIBAAKCAQEA63iS
AMPLEzUKpI8sB/AQdbZwwrDkSAMPLEt5xNM/wQrO\n5l/yWp3lCElSqnKSAMPLE8TcrkxuCyuxaV7B5
ryq3w+g3R1x\n-----END RSA PRIVATE KEY-----\n"
        }
      }
    }
  }
}
```

For more information, see *Updating an Identity Zone* in the [Cloud Foundry documentation](#).

4. The first time you set a signing key for an identity zone, existing issued tokens are immediately invalidated for online validation. Restart apps that do offline validation so that new signing keys take effect.

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

Configuring Internal User Store



Warning: Pivotal Single Sign-On v1.11 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 Platform admins can configure a Pivotal Single Sign-On service plan to manage user access to Pivotal Platform apps with the internal user store.

Overview

By default, each Single Sign-On service plan comes with an internal user store, which natively stores user accounts in a User Account and Authentication (UAA) database.

To manage the internal user store:

1. [Configure the Internal User Store](#)
2. [Add Internal Users Using UAAC](#)
3. [Test Identity Provider Configurations](#)

You can also configure a Single Sign-On service plan to use an external identity provider to manage user accounts. For more information, see [Configuring External Identity Providers](#).

Configure the Internal User Store

To configure the internal user store:

1. Log in to the **SSO Operator Dashboard** at <https://p-identity.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. Under **Name**, click the plan name and select **Manage Identity Providers** from the dropdown.
3. Under **Name**, click **Internal User Store** and select **Edit Provider** from the dropdown.
4. Under **Email Domains**, enter a comma-separated list of the email domains for service plan.

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

5. (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 admins, from managing internal users.



Note: The login page does not include **Create Account** and **Reset Password** links if you select this option.

Authentication Policy

☐ Disable Internal Authentication

☐ Disable User Management

- Under **Password Policy Settings**, select **Use Recommended Settings**, **Use Default Settings**, or enter custom settings in the **Password Complexity** and **Lockout Policy** fields.

Password Policy Settings

[Use Recommended Settings](#)

[Use Default Settings](#)

Password Complexity

Min Length

1

Min

Uppercase

0

Min

Special Characters

0

Min

Lowercase

0

Min

Numerals

0

Min

Lockout Policy

Failures Allowed

5

Max

Lockout Period

300

Seconds

Password Expires

0

Months

See the following table for configuration instructions:

Field	Instructions
Password Complexity	
Min Length	Enter the minimum password length.
Uppercase	Enter the minimum number of uppercase characters required in a password.
Lowercase	Enter the minimum number of lowercase characters required in a password.
Special Characters	Enter the minimum number of special characters required in a password.
Numerals	Enter the minimum number of numeric characters required in a password.
Lockout Policy	
Failures Allowed	Enter the number of failed login attempts permitted per hour before a user is locked out.

Lockout Period	Enter the number of seconds a user is locked out for after excessive failed login attempts.
Password Expires	Enter the number of months passwords are valid for before users need to enter a new password.

- Click **Save Identity Provider**.

Add Internal Users Using UAAC

You can create new internal user accounts with the UAA Command Line Interface (UAAC). You can also use the **Internal Users** admin pane to send invitations to users to enable them to add themselves to the internal user store. However, you cannot use the admin pane to add users directly. For information about the admin pane, see [Manage Users in an Internal User Store](#).

To create new internal user accounts with the UAAC:

- If you do not already have the UAAC installed, install the UAAC by running the following command:

```
gem install cf-uaac
```

- Create an admin client that can manage users for the Single Sign-On service plan with the following scopes:

```
✦ clients.admin
✦ scim.read
✦ scim.write
```

To create an admin client, see [Create Admin Client](#).

- Record the **App ID** and **App Secret**. These are used as your client ID and client secret.
- Target the login portal of your Single Sign-On service plan by running the following command:

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

Where `AUTH-DOMAIN` is the **Auth Domain** you entered in [Create or Edit Service Plans](#).

- Obtain an access token for your admin client by running the following command:

```
uaac token client get APP-ID
```

Where `APP-ID` is the **App ID** you recorded in the above step.

- When prompted for `Client secret`, enter the **App Secret** admin client secret you recorded in the above step.
- Add new users by running the following command:

```
uaac user add --emails USER-EMAIL
```

Where `USER-EMAIL` is the email address for the user you are creating.

8. When prompted for **User name** and **Password**, enter a username and password for the user you are creating.
9. (Optional) Create a user group and add users to the group by doing the following:
 1. Create the user group by running the following command:

```
uaac group add GROUP-NAME
```

Where **GROUP-NAME** is the name of the group you are creating.

2. Add a member to your new group by running the following command:

```
uaac member add GROUP-NAME USER-NAME
```

Test Identity Provider Configurations

Pivotal provides sample apps you can deploy to validate your identity provider configurations. To deploy a sample app, follow the instructions in [identity-sample-apps](#) in Github.

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

Configuring External Identity Providers



Warning: Pivotal Single Sign-On v1.11 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 Platform admins configure a Pivotal Single Sign-On service plan to manage user access to Pivotal Platform apps using an external identity provider (IdP).



Note: You can also configure a Single Sign-On service plan to use the internal user store to manage user accounts. For more information, see [Configuring Internal User Store](#).

Configure an External Identity Provider

You can configure Single Sign-On to use external IdPs that support:

- SAML 2.0. See [Add a SAML Provider](#).
- OpenID Connect (OIDC). See [Add an OIDC Provider](#).
- LDAP. See [Add an LDAP Provider](#).

To delete an external IdP, see [Delete an External Identity Provider](#) below.

Add a SAML Provider

To add an external SAML IdP:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your Pivotal Application Service tile in Ops Manager on the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click **New Identity Provider**.
4. Enter an **Identity Provider Name**. This name is displayed as a link on the login page.
5. Enter an **Identity Provider Description**. This is the name of the group that the IdP authenticates. This description is displayed to the Space Developers when they select an IdP for their app.
6. Under **Identity Provider type**, select **SAML 2.0**.
7. Configure **Identity Provider Metadata** by configuring one of the below fields as follows:

Identity Provider Metadata

Identity Provider Metadata URL*

[Fetch Metadata](#)

▼ SAML File Metadata (optional)

[Upload Identity Provider Metadata](#)

XML files only

Field	Instructions
Identity Provider Metadata URL	Enter a Identity Provider Metadata URL and click Fetch Metadata . If you select this option, your metadata is periodically fetched from the configured URL. This keeps the metadata up-to-date.
SAML File Metadata	Click Upload Identity Provider Metadata and upload the XML metadata that you downloaded from your external IdP. If you select this option, your metadata is not periodically updated. If the metadata changes, you must manually upload the file again.

- Enter **Email Domains**. This is a comma-separated list of domains for IdP discovery.
- (Optional) Under **Advanced Settings**, click **Attribute Mappings** and configure the fields as follows:

Advanced Settings

▼ Attribute Mappings (optional)

User Attributes

Map the incoming user attributes to known user schema.

User Schema Attribute

Attribute Name



Custom Attributes

Map additional user attributes.

☐ Persist Custom Attributes (Expose custom user attributes through the /userinfo endpoint)

Custom Attribute Name

Attribute Name



Field	Instructions
-------	--------------

User Attributes	<p>Enter any user attributes to propagate from the IdP to the service provider. These attributes can include email addresses, first names, last names, or external groups. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <ol style="list-style-type: none"> 1. Select a User Scheme Attribute from the dropdown. 2. Enter a SAML Attribute Name with the corresponding attribute from the incoming SAML assertion.
Custom Attributes	<p>Enter any custom attributes to propagate from the IdP to the service provider. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <ol style="list-style-type: none"> 1. (Optional) To expose custom user attributes through the <code>/userinfo</code> endpoint, select Persist Custom Attributes. Your app must also have the <code>user_attributes</code> scope assigned for the custom attributes to appear. 2. Enter a Custom Attribute Name. 3. Enter a SAML Attribute Name with the corresponding attribute from the incoming SAML assertion.

10. Click **Create Identity Provider** to save the IdP.
11. (Optional) Configure the service provider SAML settings for signing authentication requests and incoming assertions, by following the procedure in [Configure SAML Settings](#) below.
12. (Optional) Transfer group memberships and roles from existing SAML groups to Single Sign-On by following the procedures in [Create or Edit Resource Permissions Mapping](#) and [Configure Group Allowlist for an External Identity Provider](#) below.

Configure SAML Settings

For each plan, you can configure SAML settings when SAML is used for exchanging authentication and authorization data between the IdP and the service provider.

Single Sign-On enables the ability to sign authentication requests and require signed assertions from the external IdP.

To configure SAML settings:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click **Configure SAML Service Provider**.
4. Configure the fields as follows:

Field	Description
Perform signed authentication requests	Select this checkbox to enable the service provider to sign requests sent to the external IdP.
Require signed assertions	Select this checkbox to enable the service provider to require that responses from the external IdP are signed

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

6. Click **Download Metadata**.

Add an OIDC Provider

To add a external OIDC IdP:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager on the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click **New Identity Provider**.
4. Enter an **Identity Provider Name**. This name must match your origin key. However, you can replace lowercase letters with capital letters and replace dashes with spaces. This name is displayed as a link on the login page.
5. Enter an **Identity Provider Description**. This is the name of the group that the IdP authenticates. This description is displayed to the Space Developers when they select an IdP for their app.
6. Under **Identity Provider type**, select **OpenID Connect**.
7. Configure **OpenID Connect Settings** by configuring one of the below fields as follows: The tabs below expand to show instructions for each type of endpoint URL.

Discovery Endpoint URL	Authorization Endpoint URL		
<div> <div>OpenID Connect Settings</div> <div> <input type="checkbox"/> Skip SSL Validation <input checked="" type="checkbox"/> Enable Discovery </div> <div> Discovery Endpoint URL* <input type="text" value="https://{oidc-provider}/.well-known/openid-configuration"/> </div> <div> <input type="button" value="Fetch Scopes"/> </div> <div> Relying Party OAuth Client ID* <input type="text"/> </div> <div> Relying Party OAuth Client Secret* <input type="text"/> </div> <div> Scopes* <input type="button" value="All Selected"/> </div> </div>			
<ol style="list-style-type: none"> 1. (Optional) If you do not want to use SSL validation, select Skip SSL Validation. 2. Ensure the Enable Discovery checkbox selected. 3. Configure the fields as follows: <table> <thead> <tr> <th>Field</th> <th>Instructions</th> </tr> </thead> </table>		Field	Instructions
Field	Instructions		

Discovery Endpoint URL	Enter the discovery endpoint URL from the IdP metadata.
Relying Party OAuth Client ID	Enter the client ID from the IdP.
Relying Party OAuth Client Secret	Enter the client secret from the IdP.

- Click **Fetch Scopes**.
- Select the applicable **Scopes** for your IdP

- Enter **Email Domains**. This is a comma-separated list of domains for IdP discovery.
- (Optional) Under **Advanced Settings**, click **Attribute Mappings** and configure the fields as follows:

Advanced Settings

▼ Attribute Mappings (optional)

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

User Schema Attribute	Attribute Name
Select an attribute	Attribute Name

Custom Attributes
Map additional user attributes.

☐ Persist Custom Attributes (Expose custom user attributes through the /userinfo endpoint)

Custom Attribute Name	Attribute Name
Custom attribute name	Attribute Name

Field	Instructions
User Attributes	<p>Enter any user attributes to propagate from the IdP to the service provider. These attributes can include email addresses, first names, last names, or external groups. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <ol style="list-style-type: none"> Select a User Scheme Attribute from the dropdown. Enter a ID Token Attribute Name with the corresponding attribute from the incoming OIDC ID token

Custom Attributes	<p>Enter any custom attributes to propagate from the IdP to the service provider. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <ol style="list-style-type: none"> (Optional) To expose custom user attributes through the <code>/userinfo</code> endpoint, select Persist Custom Attributes. Your app must also have the <code>user_attributes</code> scope assigned for the custom attributes to appear. Enter a Custom Attribute Name. Enter a ID Token Attribute Name with the corresponding attribute from the incoming OIDC ID token.
--------------------------	---

- Click **Create Identity Provider** to save the IdP.
- (Optional) Transfer group memberships and roles from existing OIDC groups to Single Sign-On by following the procedures in [Create or Edit Resource Permissions Mapping](#) and [Configure Group Allowlist for an External Identity Provider](#) below.

Add an LDAP Provider

When integrating Single Sign-On with a LDAP external IdP, authentication is chained. An authentication attempt with user credentials is first attempted against the internal user store before the external LDAP IdP.

When using an LDAP external IdP, you should not:

- Bootstrap or Create Users in the UAA Database:** This can cause user collisions.
- Enable Manual Lockouts:** For example, lockouts that result from users using the same account.
- Enable Automated Deletions:** This can disrupt service accounts and prevent user logins.

VMware recommends that you do not reuse LDAP service accounts across environments. You can only have one LDAP external IdP per service plan.

To add a LDAP IdP:

- Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your Pivotal Application Service tile in Ops Manager on the **Credentials** tab.
- Click the plan name and select **Manage Identity Providers** from the dropdown.
- Click **New Identity Provider**.
- Enter an **Identity Provider Name**. This name is displayed as a link on the login page.
- Enter an **Identity Provider Description**. This is the name of the group that the IdP authenticates. This description is displayed to the Space Developers when they select an IdP for their app.
- Under **Identity Provider type**, select **LDAP**. You can only have one LDAP provider per Service Plan.
- Configure the fields as follows:

Connection

Hostname*

localhost

Port*

389

Security protocol

None

Referral

follow

User DN*

cn=admin,ou=Users,dc=test,dc=com

Bind Password*

Users

Search Base*

dc=test,dc=com

Search Filter

cn={0}

☒ Just in Time Provisioning

Groups

Search Base

ou=scopes,dc=test,dc=com

Search filter

member={0}

Field	Instructions
Connection	
Hostname	Enter the hostname for your LDAP server.
Port	Enter the post name for your LDAP server.
Security protocol	Select the security protocol that your LDAP uses for connection.
Referral	Select how UAA handles LDAP server referrals to other user stores.
User DN	Enter the LDAP Distinguished Name (DN) for binding to your LDAP server.
Bind Password	Enter the password for binding to your LDAP server.
Users	
Search Base	Enter the location in the LDAP directory tree where LDAP user search begins. The LDAP search base typically matches your domain name.

Search Filter (Optional)	Enter a string to use for LDAP user search criteria.
Just in Time Provisioning	If this option is enabled, users are created at login time. If this option is not enabled, users must be created before being able to log in
Groups	
Search Base (Optional)	Enter the location in the LDAP directory tree where the LDAP group search begins. To use the memberOf attribute on user objects, enter the value <code>memberOf</code> as the Search Base instead of an LDAP path for a group organizational unit. This causes Single Sign-On to ignore the Search Filter value.
Search filter (Optional)	Enter a string that defines LDAP group search criteria. The standard value is <code>member={0}</code> .

- Enter **Email Domains**. This is a comma-separated list of domains for IdP discovery.
- (Optional) Under **Advanced Settings**, click **Attribute Mappings** and configure the fields as follows:

Advanced Settings

▼ Attribute Mappings (optional)

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

User Schema Attribute	Attribute Name
Select an attribute	Attribute Name

Custom Attributes
Map additional user attributes.

☐ Persist Custom Attributes (Expose custom user attributes through the /userinfo endpoint)

Custom Attribute Name	Attribute Name
Custom attribute name	Attribute Name

Field	Instructions
User Attributes	<p>Enter any user attributes to propagate from the IdP to the service provider. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <p>The only configurable attributes for LDAP are <code>given_name</code>, <code>family_name</code>, and <code>phone_number</code>. Configuring <code>email</code> and <code>external_groups</code> has no effect on your LDAP integration.</p> <ol style="list-style-type: none"> Select a User Scheme Attribute from the dropdown. Enter a LDAP Attribute Name with the corresponding attribute LDAP.

Custom Attributes	<p>Enter any custom attributes to propagate from the IdP to the service provider. They are sent to apps through OpenID tokens, along with any other stored user information issued by Single Sign-On.</p> <ol style="list-style-type: none"> (Optional) To expose custom user attributes through the <code>/userinfo</code> endpoint, select Persist Custom Attributes. Your app must also have the <code>user_attributes</code> scope assigned for the custom attributes to appear. Enter a Custom Attribute Name. Enter a LDAP Attribute Name with the corresponding attribute from LDAP.
--------------------------	---

- Click **Create Identity Provider** to save the IdP.
- (Optional) Transfer group memberships and roles from existing SAML groups to Single Sign-On by following the procedures in [Create or Edit Resource Permissions Mapping](#) and [Configure Group Allowlist for an External Identity Provider](#) below.

Delete an External Identity Provider



Note: Deleting an external IdP deletes all of its configurations. This prevents users from authenticating through the external IdP. You cannot undo this action.

To delete an external IdP:

- Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager on the **Credentials** tab.
- Click the plan name and select **Manage Identity Providers** from the dropdown.
- Click on the name of your external IdP.
- Click **Delete** at the bottom of the page.
- In the dialog box that appears, click **Delete Identity Provider** to confirm that you want to delete the IdP, along with all of its configurations.

Transfer Group Membership and Roles

You can use the SSO Operator Dashboard to configure resource permissions mappings and group allowlists to transfer group membership and roles from existing IdP groups to Single Sign-On.

SSO Operator Dashboard enables you to:

- [Create or Edit Resource Permissions Mapping](#)
- [Delete Resource Permissions Mapping](#)
- [Configure Group Allowlist for an External Identity Provider](#)

Create or Edit Resource Permissions Mapping

After a Space Developer defines the resources required by an app, an admin can map existing groups to those resources. For information about how Space Developers define resources, see [Create or Edit Resources](#).

After resource permissions mappings are configured and a user authenticates, the user group memberships are mapped to scopes in the resulting token.

To create or edit resource permissions mappings:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager on the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click the name of the external IdP you want to define permissions for and select **Resource Permissions** from the dropdown.
4. Click **New Permissions Mapping**.
5. Enter a **Group Name**. The group name depends on the IdP you are configuring:
 - ✦ **SAML or OIDC providers:** Enter the group name that you defined in the external IdP.
 - ✦ **LDAP providers:** Enter the fully qualified LDAP paths. The path uses this format: `cn=XXX-group,ou=users,o=YYY,dc=ZZZ,dc=com`.
6. Click **Select Permissions** and select 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.



Note: You can use the UAA API to automate the above procedure. To do this automation, you need an identity zone admin client. For instructions about creating the identity zone admin client, see [Create a UAA Identity Zone Admin Client](#).

For instructions about granting admin permissions to mapped external identity groups, see [Grant Admin Permissions to an External Group \(SAML or LDAP\)](#) in the Cloud Foundry documentation.

Delete Resource Permissions Mapping

To delete a resource permissions mapping:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager on the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click on the name of the external IdP you want to delete permissions for and select **Resource Permissions** from the dropdown.
4. Click the group name of the resource permission you want to delete.
5. Click **Delete** at the bottom of the page.

- On the dialog box, click **Delete Permissions Mapping** to delete the resource.



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



Note: You can use the UAA API to automate the above procedure. To do this automation, you need an identity zone admin client. For instructions about creating the identity zone admin client, see [Create a UAA Identity Zone Admin Client](#).

For information about un-mapping a group mapping, see [Unmap](#) in the UAA API documentation.

Configure Group Allowlist for an External Identity Provider

An admin can include groups from an external IdP in a group allowlist. The list of groups in the allowlist propagates in the ID token when a user authenticates using an external IdP.

An app can retrieve from the ID token the list of external groups that the user belongs to. An admin can use these groups to assign permissions by group rather than by individual users.

For information about how to create resource permission mappings, see [Create or Edit Resource Permissions Mapping](#) above.



Note: For an app to retrieve an ID token or a `/userinfo` response containing external groups, the app must request the `roles` claim and the group allowlist must include the external groups.

To configure a group allowlist:

- Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your UAA admin credentials. You can find these credentials in your PAS tile in Ops Manager on the Credentials tab.
- Click the plan name and select **Manage Identity Providers** from the dropdown.
- Click on the name of your external IdP and select **Group Whitelist** from the dropdown.
- Enter a group name from your external IdP. You can use a regex to add group names. For example, you can use `*` to refer to all groups.
- Click the **+** icon.
- Click **Save Group Whitelist**.

Test Identity Provider Configurations

Pivotal provides sample apps you can deploy to validate your IdP configurations.

To deploy a sample app, see [identity-sample-apps](#) in GitHub.

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

Updating Identity Providers with UAAC



Warning: Pivotal Single Sign-On v1.11 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 update the configuration of identity providers using the User Account and Authentication Command Line Interface (UAAC).

For instructions on configuring identity providers, see [Configuring Identity Providers](#).

Create a UAA Identity Zone Admin Client

To use the UAAC with your Pivotal Single Sign-On 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 Single Sign-On service plan.

Create an Admin Client

To create a UAA admin client:

1. Follow the procedure in [Create an Admin Client](#).
2. Using the instructions above, give the admin client the `idps.read`, `idps.write` and, `clients.admin` scopes.
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 display your client context and verify that you have `idps.read`, `idps.write`, and `clients.admin` under the scope section.

```
uaac context
```

For example:

```
$ uaac context
[1]*[ExampleAppID]
  client_id: ExampleAppID
  access_token: aBcdEfg0hIJKlm123.e
  token_type: bearer
  expires_in: 43200
  scope: uaa.resource idps.read idps.write clients.admin
  jti: 91b3-abcd1233
```

5. 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 Operator Dashboard after you create the identity zone client.

6. 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: *****
```

7. 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: asdioqwue1k12312.e21e
  token_type: bearer
  expires_in: 43200
  scope: uaa.admin
  jti: 123908dk11-23298
```

You can now do operator level API configurations for the Single Sign- On service plan. You do not have permissions for any other Single Sign- On service plan.

Update UAA Identity Provider Configurations with the API

This section shows how to use the UAAC to update UAA identity provider 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 the `PUT` command. You must also provide all configuration values, otherwise, data can be lost.

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

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

1. Create an identity zone admin client following [Create a UAA Identity Zone Admin Client](#) above.
2. To retrieve your identity provider ID, run the following command:

```
uaac curl -k /identity-providers > ID-TEXT.txt
```

Where:

- ✦ `YOUR-IDENTITY-PROVIDER-ID` is your identity provider ID.
- ✦ `ID-TEXT.txt` is the name of your text file.

The command above outputs a JSON blob similar to the example below.

```
{
  "type" : "uaa",
  "config" : "null",
  "id" : "4be9d903-b6ce-4a57-9c56-94e731b58628",
  "originKey" : "uaa",
  "name" : "uaa",
  "version" : 3,
  "created" : 946684800000,
  "last_modified" : 1538589026045,
  "active" : true,
  "identityZoneId" : "uaa"
```

Your identity provider ID is the value of `id`. In most cases, this command returns one identity provider. If there are several, you can identify your identity provider by the `name`.

- Run the following command, directing the output to a text file:

```
uaac curl -k /identity-providers/YOUR-IDENTITY-PROVIDER-ID > JSON-BLOB.txt
```

Where:

- ✦ `YOUR-IDENTITY-PROVIDER-ID` is the identity provider ID retrieved in the above step.
- ✦ `JSON-BLOB.txt` is the name of your text file.

- The command above outputs a JSON blob similar to the example below. Confirm that the ID in this output matches `YOUR-IDENTITY-PROVIDER-ID`.

```
{
  "type": "uaa",
  "config": "{ \"emailDomain\":null,
  \"additionalConfiguration\":null,
  \"providerDescription\":null,
  \"passwordPolicy\":null,
  \"lockoutPolicy\":null,
  \"disableInternalUserManagement\":false}",
  "id": "b38dfbbc-f187-4eeb-a3f3-21a3c72c6975",
  "originKey": "uaa",
  "name": "uaa",
  "version": 0,
  "created": 1530220213000,
  "last_modified": 1530220213000,
  "active": true,
  "identityZoneId": "uaa"
}
```

- 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.

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



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

```
$ uaac curl -k /identity-providers/YOUR-IDENTITY-PROVIDER-ID -X PUT \
-H 'Content-Type: application/json' -d "$(cat file.txt)"
```

Where:

- `YOUR-IDENTITY-PROVIDER-ID` is the identity provider ID retrieved in the above step.

A minimal example command would look similar to the following:

```
$ uaac curl -k /identity-providers/b38dfbbc-f187-4eeb-a3f3-21a3c72c6975 \
-X PUT \
-H 'Content-Type: application/json' \
-d '{
  "type": "uaa",
  "config": {
    "emailDomain": null,
    "providerDescription": null,
    "passwordPolicy": null,
    "lockoutPolicy": {
      "lockoutPeriodSeconds": 8,
      "lockoutAfterFailures": 8,
      "countFailuresWithin": 8
    },
    "disableInternalUserManagement": false
  },
  "originKey": "uaa",
  "name": "uaa",
  "version": 0,
  "active": true
}'
```

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

Enable Client Auth for OpenID Connect (OIDC)

Some OIDC providers only support client secrets via POST instead of via basic authentication.



Note: Azure Active Directory integrations using the response type of code only work with `clientAuthInBody`.

To enable client authentication via POST, do the following:

- Follow the steps 1-4 in [Update UAA Identity Provider Configurations with the API](#) above.
- In step 5, add the following line to the config section in the JSON blob:

```
"clientAuthInBody": true
```

3. Complete the procedure by following step 6.

For more information about client authentication, see [OAuth/OIDC](#) in the UAA API documentation.

Enable Password Grant for OpenID Connect (OIDC)

You can enable OIDC password grant to permit native apps to forward credentials to an enterprise identity provider for authentication.

To enable OIDC password grant, do the following:

1. Follow the steps 1-4 in [Update UAA Identity Provider Configurations with the API](#) above.
2. In step 5, add the following line to the config section in the JSON blob:

```
"passwordGrantEnabled": true
```

3. Complete the procedure by following step 6.

For more information about OIDC password grant, see [OAuth/OIDC](#) in the UAA API documentation.

Skip SSL Validation for SAML

For cases where the SAML provider is configured using a valid SAML URL and a self-signed certificate, you might need to skip SSL validation.



Note: This section assumes you have created the SAML identity provider, most likely by providing the SAML metadata directly. For more information, see [Add a SAML Provider](#).

To enable skipping SSL validation, do the following:

1. Follow the steps 1-4 in [Update UAA Identity Provider Configurations with the API](#) above.
2. In step 5, add the following line to the config section in the JSON blob:

```
"skipSslValidation": true
```

3. Complete the procedure by following step 6.

For more information about skipping SSL validation, see [SAML](#) in the UAA API documentation.

To test your identity provider configurations, you can deploy the [Pivotal Single Sign-On Sample Applications](#) to validate the effects of the instructions given in this guide.

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

Managing Clients with UAAC



Warning: Pivotal Single Sign-On v1.11 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 plan administrators use the User Account and Authentication Command Line Interface (UAAC) to manage existing UAA Identity Zone clients.

About Managing Clients with UAAC

This section explains when and why you use the UAAC to update UAA Identity Zone clients.

All clients mentioned on this page are UAA Identity Zone clients. However, there are two kinds of UAA Identity Zone clients:

- **Non-Admin clients**—When app developers configure their apps to use Pivotal Single Sign-On, each app corresponds to a non-admin client for a Single Sign-On service plan.
- **Admin clients**—These can modify other clients and are created by completing the procedure below. See [Create an Admin Client](#).

When Not to Use UAAC

Do not use the UAAC to do the following:

- Create clients—Do not create clients through UAAC because additional metadata is required for their usage by Single Sign-On.
- Make most types of updates—Most updates for UAA Identity Zone clients can be made through the SSO Developer Dashboard.

When to Use UAAC

Some updates cannot be done through the SSO Developer Dashboard and so must be made through the UAAC. You need to use the UAAC if you want to set a configuration to a value that is not listed on the SSO Developer Dashboard.

Create an Admin Client

To use the UAAC to modify clients, you need an admin client that corresponds to your Single Sign-On service plan.

If you do not already have an admin client for your UAA Identity Zone, follow the steps below to

create an admin client.



Note: You can use the same admin client for updating service plans and identity providers. For information, see [Updating Service Plans with UAAC](#) and [Updating Identity Providers with UAAC](#).

1. Target your Pivotal Platform deployment using `cf`.
2. Target an org and space that your service plan is visible in.
3. If you have not already created a service instance for your service plan, create one now. For how to create an instance, see [Create a Service Instance](#). The service instance exposes the SSO Developer Dashboard.
4. Log in to the SSO Developer Dashboard as an administrator. You can find the dashboard URL by using Apps Manager or `cf service SERVICE-INSTANCE-NAME`.
5. Click **New App**.
6. Enter an **App Name**.
7. Under **Select an Application Type**, select **Service-to-Service App**.
8. Click **Select Scopes > Admin Permissions**.

Set the scopes as necessary for configuring the UAA resource.

For...	Add these scopes...	For more information, see...
updating UAA clients	<code>clients.admin</code>	Update Clients with UAAC below .
managing Single Sign- On service plans	<code>clients.admin</code>	Updating Service Plans with UAAC .
updating identity providers	<code>idps.read</code> and <code>idps.write</code>	Updating Identity Providers with UAAC .

9. Record the **App ID** and **App Secret**.

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

Enabling Identity Provider Discovery



Warning: Pivotal Single Sign-On v1.11 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 Platform apps that use Pivotal Single Sign-On.

What IdP Discovery Does

If users with different email domains access the same Pivotal Platform app, you can configure Single Sign-On 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 where some users log in with `@example.com` username and some with `@gmail.com` username. With IdP Discovery, users with both email domains can log in from the same login 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 Administrator or a Plan Administrator.

1. Enable IdP Discovery for the Single Sign-On service instance that your app is bound to:
2. Log into the Pivotal Platform at <https://p-identity.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.
 1. Click the plan name and select **Configure** under the plan menu.
 2. Select the checkbox under the **Identity Provider Discovery** section and click **Save**.

The screenshot shows the 'Configure' page for a plan named 'Acme INC'. The page has a dark header with a 'P' logo, 'Single Sign-On | Operator Dashboard', 'Help' with an external link icon, and a user dropdown 'admin'. A breadcrumb trail reads 'Plans > Acme INC > Configure'. The main content area is titled 'Configure' and contains three sections: 'Token Policy', 'Refresh Token Expiration', and 'Identity Provider Discovery'. Under 'Token Policy', there is an 'Access Token Expiration' input field (empty) with a 'Seconds' label and a checked checkbox for 'Use System Default (12 hours)'. Under 'Refresh Token Expiration', there is a 'Refresh Token Expiration' input field (empty) with a 'Seconds' label and a checked checkbox for 'Use System Default (30 days)'. Under 'Identity Provider Discovery', there is a checked checkbox for 'Enabled'. At the bottom right are 'Cancel' and 'Save' buttons.

Single Sign-On | Operator Dashboard Help admin

Plans > Acme INC > Configure

Configure

Token Policy

Access Token Expiration Seconds

☒ Use System Default (12 hours)

Refresh Token Expiration

Refresh Token Expiration Seconds

☒ Use System Default (30 days)

Identity Provider Discovery

☒ Enabled

Cancel Save

3. Click the plan name and select **Manage Identity Providers** under the plan menu.
 4. Enter the Email domains you want to include as a comma-separated list under the configuration page for the identity provider plan.
-

Single Sign-On | Operator Dashboard Help admin

Plans > Acme INC > Identity Providers > Okta

Okta

Cancel Save Identity Provider

Identity Provider Name*

Okta

This name will show as a link on the login page

Identity Provider Description

Allows Okta to authenticate.

Identity Provider type*

SAML 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

Fetch Metadata

► SAML File Metadata (optional)

IDP Metadata

Entity ID	
SSO URL	
Name ID	

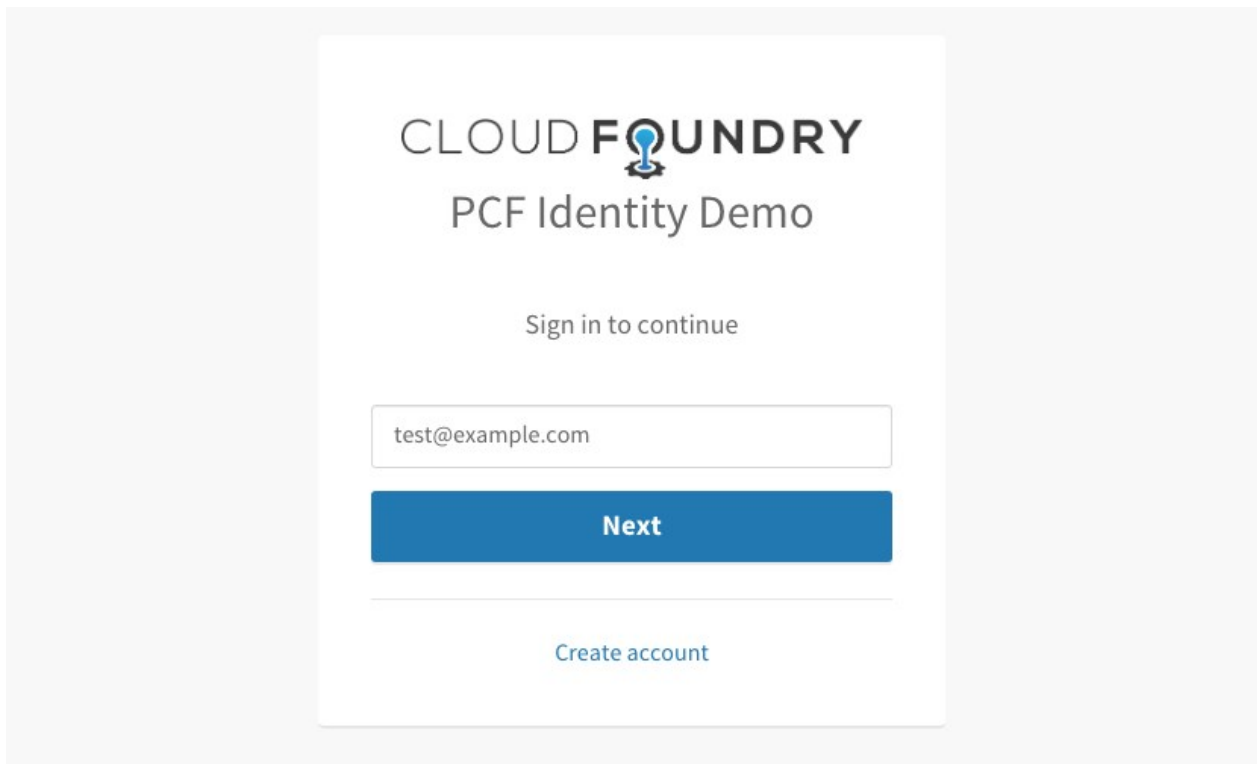
Email Domains

Provide comma-separated list of domains for identity provider discovery

example.com

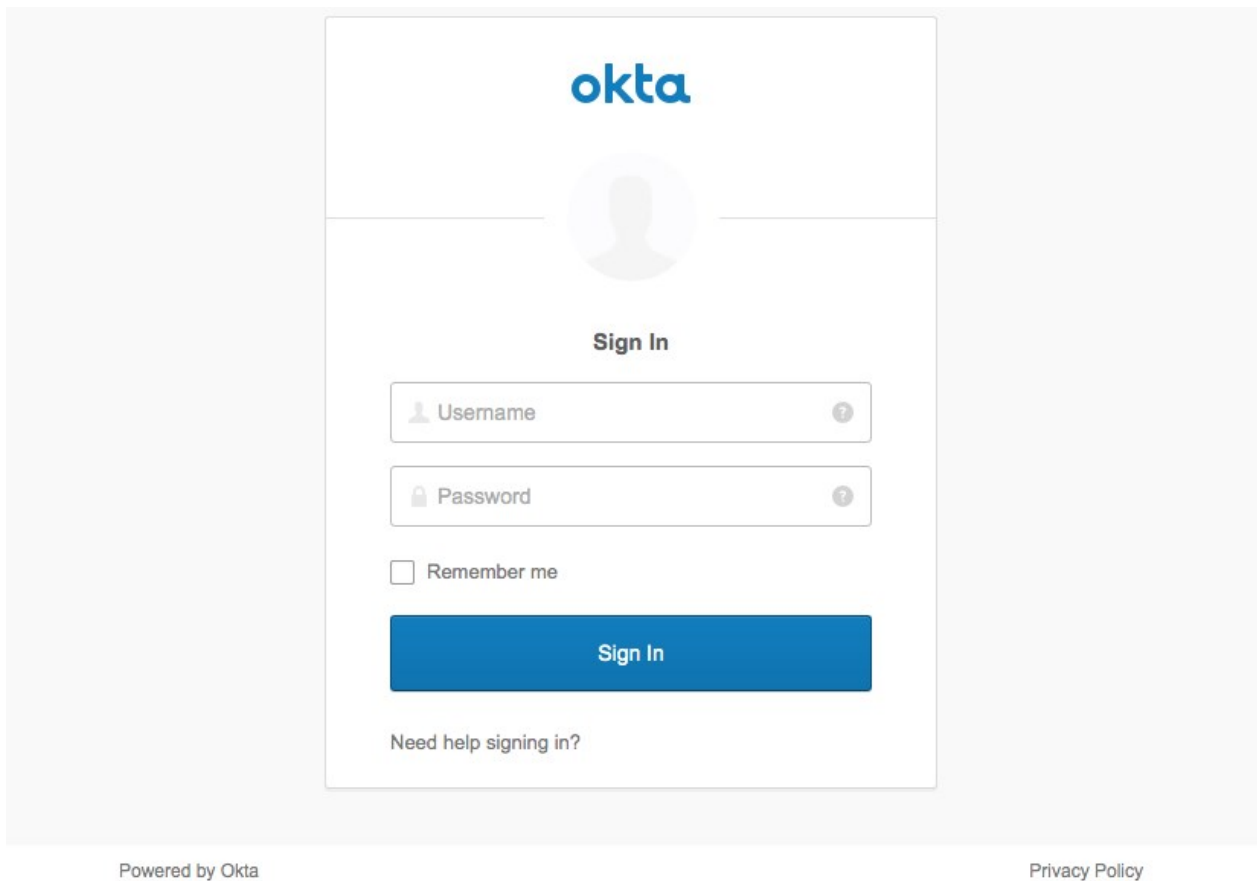
5. In Apps Manager, navigate to your space, open the **Service** tab, and select your service instance.
6. Click the **Manage** link under the service name, and edit the app configuration by selecting the required Identity Providers.

After these steps, the login page prompts for the username first:



The image shows a login page for Cloud Foundry PCF Identity Demo. At the top, the Cloud Foundry logo is displayed, followed by the text "PCF Identity Demo". Below this, the instruction "Sign in to continue" is shown. A text input field contains the email address "test@example.com". A blue button labeled "Next" is positioned below the input field. At the bottom, there is a link that says "Create account".

If the user enters their `test@example.com` username, they are redirected to the Okta login page:



The image shows the Okta login page. At the top, the Okta logo is displayed. Below the logo is a placeholder for a user profile picture. The text "Sign In" is centered below the profile picture. There are two input fields: "Username" and "Password", each with a question mark icon to its right. Below these fields is a checkbox labeled "Remember me". A blue button labeled "Sign In" is positioned below the checkbox. At the bottom, there is a link that says "Need help signing in?". At the very bottom of the page, there are two links: "Powered by Okta" on the left and "Privacy Policy" on the right.

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

Managing Users



Warning: Pivotal Single Sign-On v1.11 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 Platform Plan Administrator uses Pivotal Single Sign-On to manage user access to Pivotal Platform apps, for users with accounts in the internal user store or with external identity providers.

Manage Users in an Internal User Store

Pivotal Single Sign-On has an **Internal Users** admin pane that lets you manage user accounts in the Pivotal Platform 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 Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your User Account and Authentication (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 dropdown.
3. Click **Internal User Store** and select **Internal Users** from the dropdown. This brings you to the admin screen.

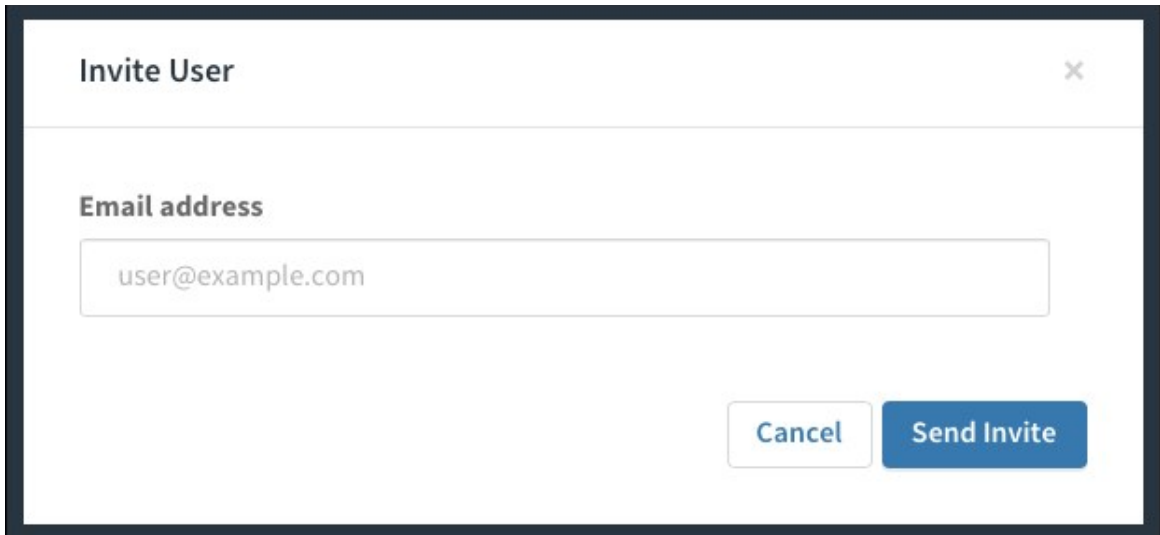
Internal Users [Invite User](#)

Search for Users

[Search](#)

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

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



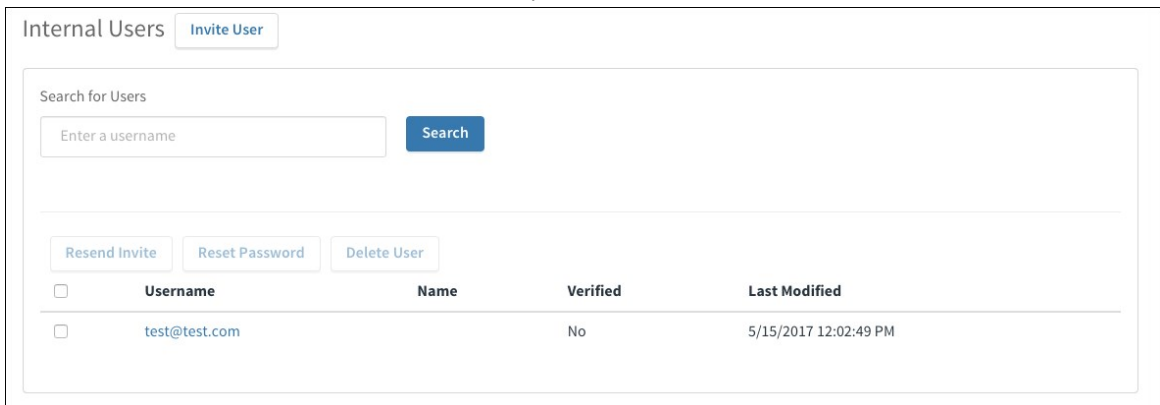
Invite User [X]

Email address

user@example.com

Cancel 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]

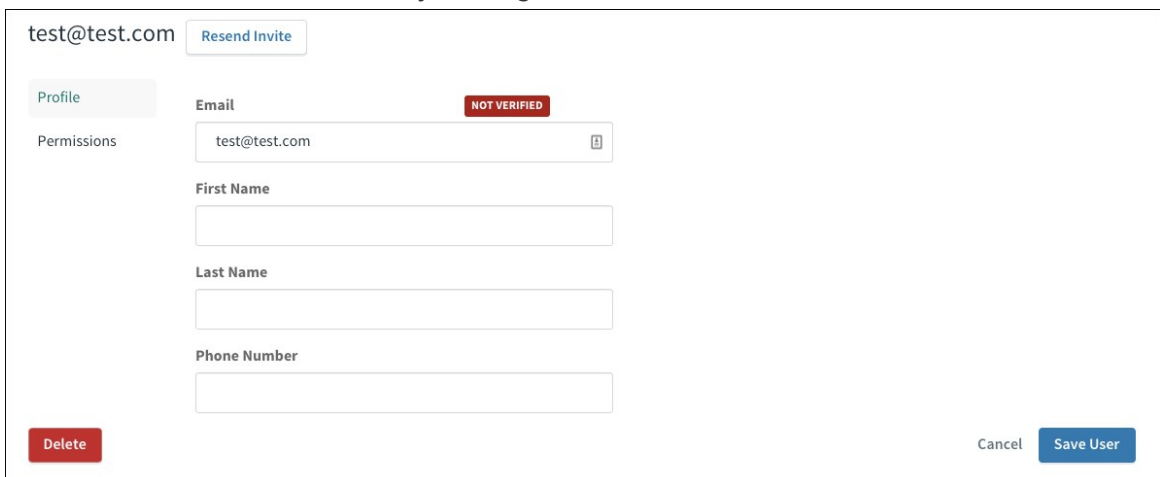
Search for Users

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.

- **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 Single Sign-On connects to, a users admin pane lets you browse, delete, and update Pivotal Platform permissions for user accounts from external identity providers. For example, **Okta SSO Users**.

To open the external identity provider users admin pane:

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> using your User Account and Authentication (UAA) admin credentials. You can find these credentials in your tile in Ops Manager under the **Credentials** tab.
2. Click the plan name and select **Manage Identity Providers** from the dropdown.
3. Click the external identity provider you want to manage and select the **Users** choice for the provider from the dropdown. This brings you to the users admin pane.

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

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

Username	Name	Verified	Last Modified
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.

tiwang+test@pivotal.io

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 with the UAA CLI

You may also use the UAA CLI, or [UAAC](#), to manage users for Single Sign-On. 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 admin for the service plan.



Note: Clients and Groups for Single Sign-On should be created directly using the SSO Operator Dashboard or through app manifest bootstrapping. Do not create these through UAAC, as additional metadata is required for their usage by Single Sign-On.

1. Install the UAA CLI, `uaac`, by running:

```
gem install cf-uaac
```

2. Target your service plan by running:

```
uaac target AUTH-DOMAIN.login.SYSTEM-DOMAIN
```

Where `AUTH-DOMAIN` is the **Auth Domain** you entered in [Create or Edit Service Plans](#).

For example:

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

- Record the **App ID** and **App Secret** from your admin client created by following the steps in [Create Admin Client](#). You must 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.
- Authenticate and obtain an access token for the admin client for your service plan by running:

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

Where:

- ✦ `ADMIN-APP-ID` is your **App ID** and
- ✦ `ADMIN-APP-SECRET` is your **App Secret**.

For example:

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

UAAC stores the token in `~/.uaac.yml`.

- Display the client context by running the following command and verify that you have the sufficient `scim.write` or `scim.create` permissions under the `scope` section:

```
uaac context
```

For example:

```
$ 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
```

- Create a new internal user by running:

```
uaac user add NEW-USERNAME -p NEW-PASSWORD --emails NEW-EMAIL
```

Replace `NEW-USERNAME`, `NEW-PASSWORD`, and `NEW-EMAIL` with appropriate information.

For example:

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

- Add any group to any user (internal or external) by running:


```
uaac member add GROUP USERNAME
```

Replace **GROUP** and **USERNAME** with appropriate information.

For example:

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

8. Delete any group from to any user (internal or external).

```
uaac member delete GROUP USERNAME
```

Replace **GROUP** and **USERNAME** with appropriate information.

For example:

```
$ 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On for Pivotal Platform.

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.

Single Sign-On 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
UAA scaling indicators	UAA VM CPU Utilization

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

Monitoring Service Plans and Apps



Warning: Pivotal Single Sign-On v1.11 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 monitor Pivotal Single Sign-On for Pivotal Platform service plans and apps.

Overview

Single Sign-On uses the User Account and Authentication (UAA) service to log security events through Loggregator. UAA security events can be filtered to destinations through a syslog drain. To configure logs to monitor Single Sign-On plan events, app, and UAA client events you need to obtain the IDs for the corresponding plan or app.

To obtain the identity zone ID for Single Sign-On plans, do one of the procedures in [Monitor Single Sign-On Plan Events](#) below.

To obtain the client ID for an app or UAA client, do the procedure in [Monitor App Events](#) below.

For information about configuring logging in Pivotal Application Service, see [Configuring Logging in Pivotal Application Service](#).

For information about UAA security events, see [UAA Logging](#).

Monitor Single Sign-On Plan Events

All Single Sign-On service plans are given a unique identity zone ID. You can monitor all events for a plan by filtering UAA generated logs using the plan's identity zone ID.

You can obtain a list of plans and their corresponding identity zone IDs by doing one of the following:

- [Use the Single Sign-On API](#), as detailed below.
- [Use the SSO Operator Dashboard](#), as detailed below.

Prerequisites

Before you can use the Single Sign-On API to monitor plan events, you must:

1. [Create an Admin Client](#).
2. [Create a UAA Identity Zone Admin Client](#).

Use the Single Sign- On API

To use the Single Sign- On API to obtain plan identity zone IDs, run the following command:

```
curl -X GET "https://sso-api.SYSTEM-DOMAIN/v1/plans" \
-H "Authorization: Bearer TOKEN"
```

Where **TOKEN** is the access token you obtained in [Create a UAA Identity Zone Admin Client](#).

For more information, see [Single Sign- On Service Plan Automation API](#) in the Single Sign- On API documentation.

Use the SSO Operator Dashboard

To use the SSO Operator Dashboard to obtain plan identity zone IDs:

1. Log into the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN/dashboard>
2. Click the plan you want to obtain the identity zone ID for and select **Edit Plan**.
3. Record the identity zone ID for your plan from the SSO Operator Dashboard URL. The URL looks similar to the one below.

```
https://p-identity.SYSTEM-DOMAIN/dashboard/edit_plan/IDENTITY-ZONE-ID
```

Where **IDENTITY-ZONE-ID** is your plan' s identity zone ID.

Monitor App Events





All apps that use Single Sign- On have a unique client ID. You can monitor app and UAA client events using the client ID.

To find your app' s client ID:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click the **Single Sign- On** service.
4. Click **Manage** next to your Single Sign- On service instance to launch the SSO Developer Dashboard.
5. Under **App**, click **Credentials** near the name of your app.
6. Record the value of **App ID**.

Service-to-Service App

my-app

App ID Unique identifier for the application	<input type="text"/>	
App Secret Authenticates the application	<input type="password"/> The App Secret can only be displayed once. Regenerate app secret	
SSO Service URL Auth domain for single sign-on	<input type="text"/>	
OAuth Token URL Client retrieves token from this endpoint	<input type="text"/>	
Token Verification Keys An endpoint which returns JWT verification keys	<input type="text"/>	

[View a larger version of this image.](#)

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

Backing Up and Restoring



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On for Pivotal Platform.

To back up and restore Single Sign-On, use the BOSH Backup and Restore (BBR) tool to back up and restore Pivotal Platform. For more information, see [Backing Up and Restoring Pivotal Platform](#).

As part of backing up and restoring Pivotal Platform, BBR also backs up and restores the Single Sign-On tile information. The information is included in the Cloud Controller (CC) and User Account and Authentication (UAA) data of Pivotal Application Service.

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

Determining Pivotal Single Sign- On App Type



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign- On app type.

Determine Your Single Sign- On App Type

Before you bind or register an app, you must determine its Single Sign- On app type and the corresponding OAuth grant type. OAuth grant types determine how the app communicates with Single Sign- On to acquire tokens for authentication and authorization purposes.

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

See the table below to determine your app's Single Sign- On app 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 App	Resource Owner Password
Single-Page JavaScript	Single-Page JavaScript App	Implicit
Service-to-Service	Service-to-Service App	Client Credentials
Web + Service-to-Service	Web + Service-to-Service App	Authorization Code and Client Credentials
Resource Server	Secured API, Database Server	n/a

The [Single Sign- On Service Sample Applications](#) GitHub repository provides examples for a few of the Single Sign- On app types listed above.



Note Pivotal recommends only using the Native app type for highly-trusted apps, such as company-owned and managed apps. The Native app type only works 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: Pivotal Single Sign-On v1.11 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.

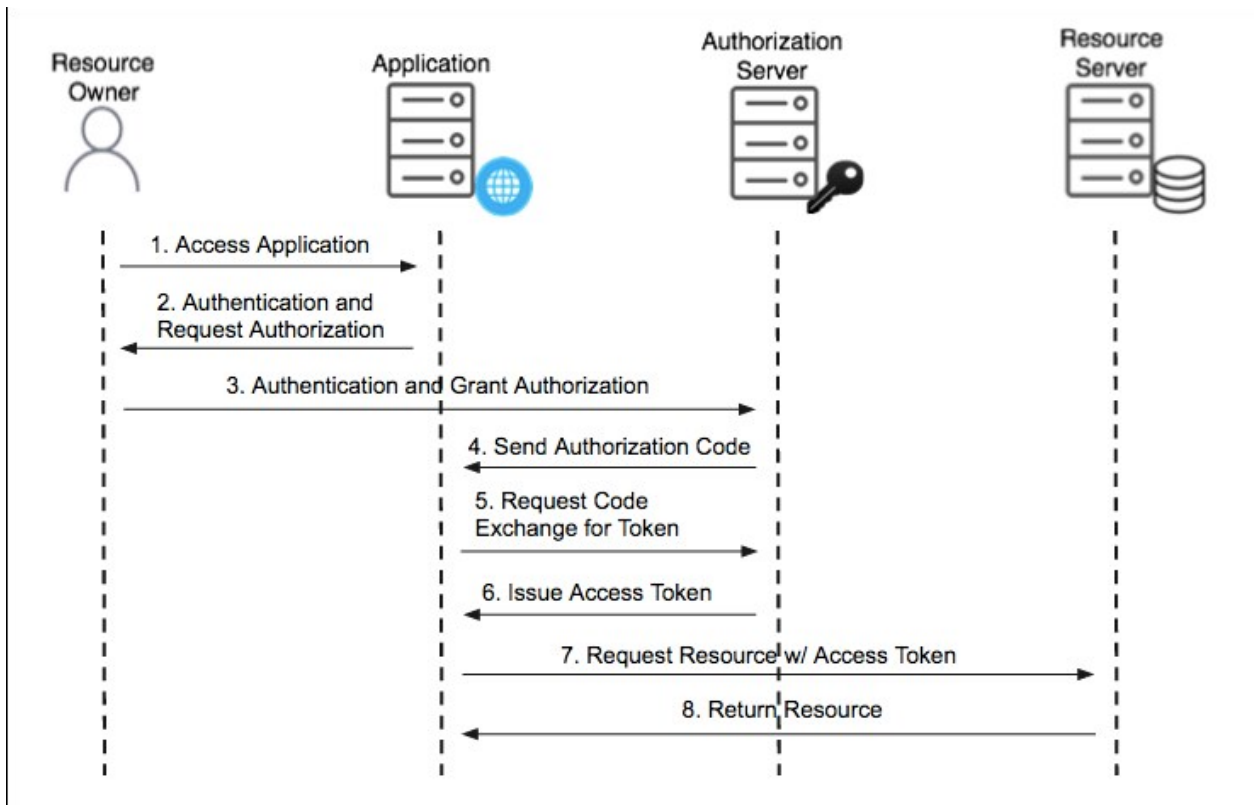
Authorization Code Grant Type Roles

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 Platform 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.

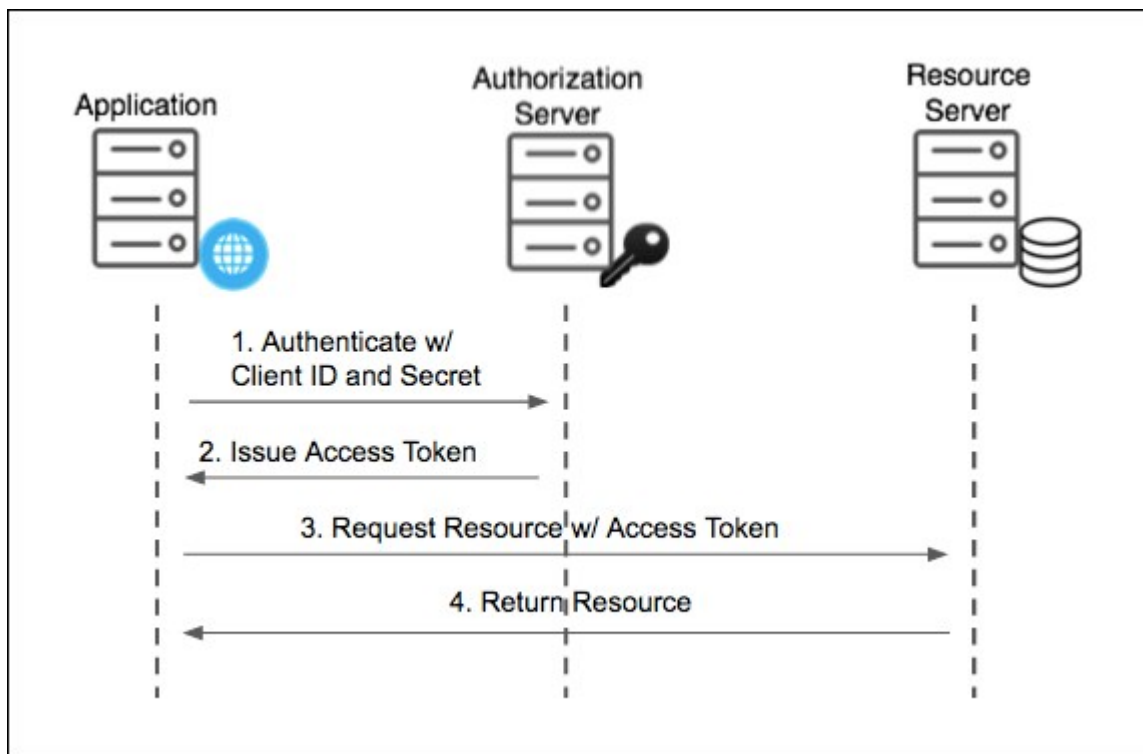
Client Credentials Grant Type Roles

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, Pivotal Single Sign-On 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.

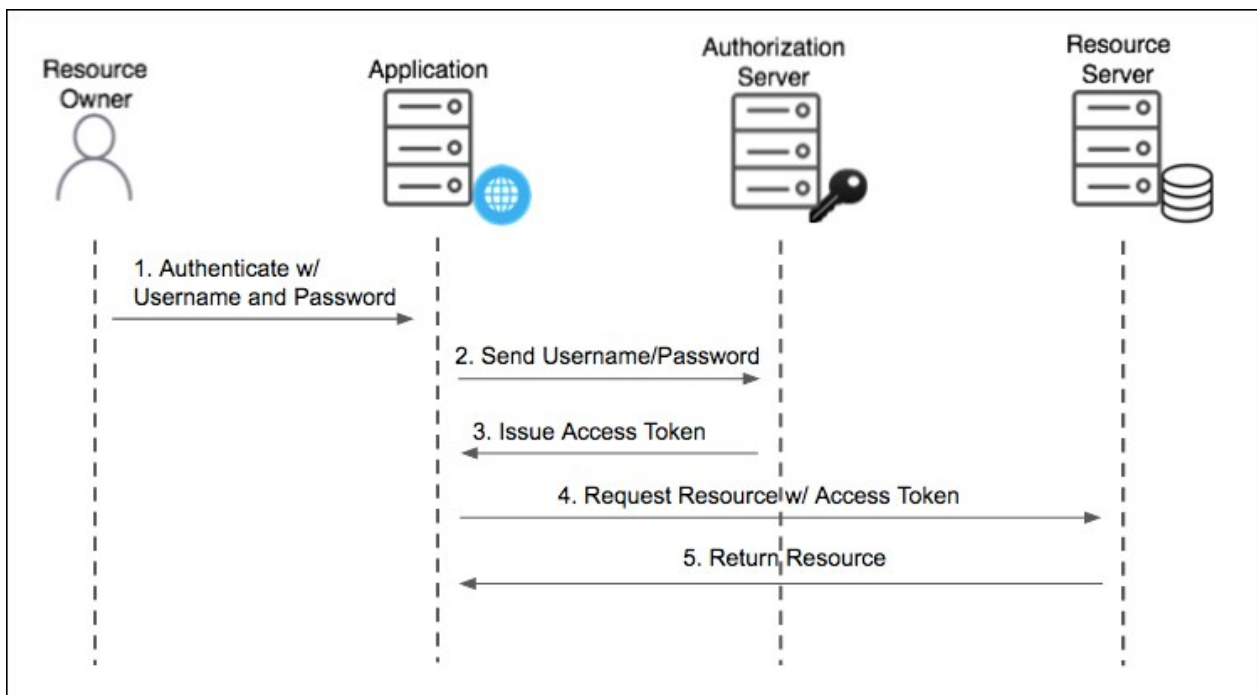
Resource Owner Password Grant Type Roles

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 Platform 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.

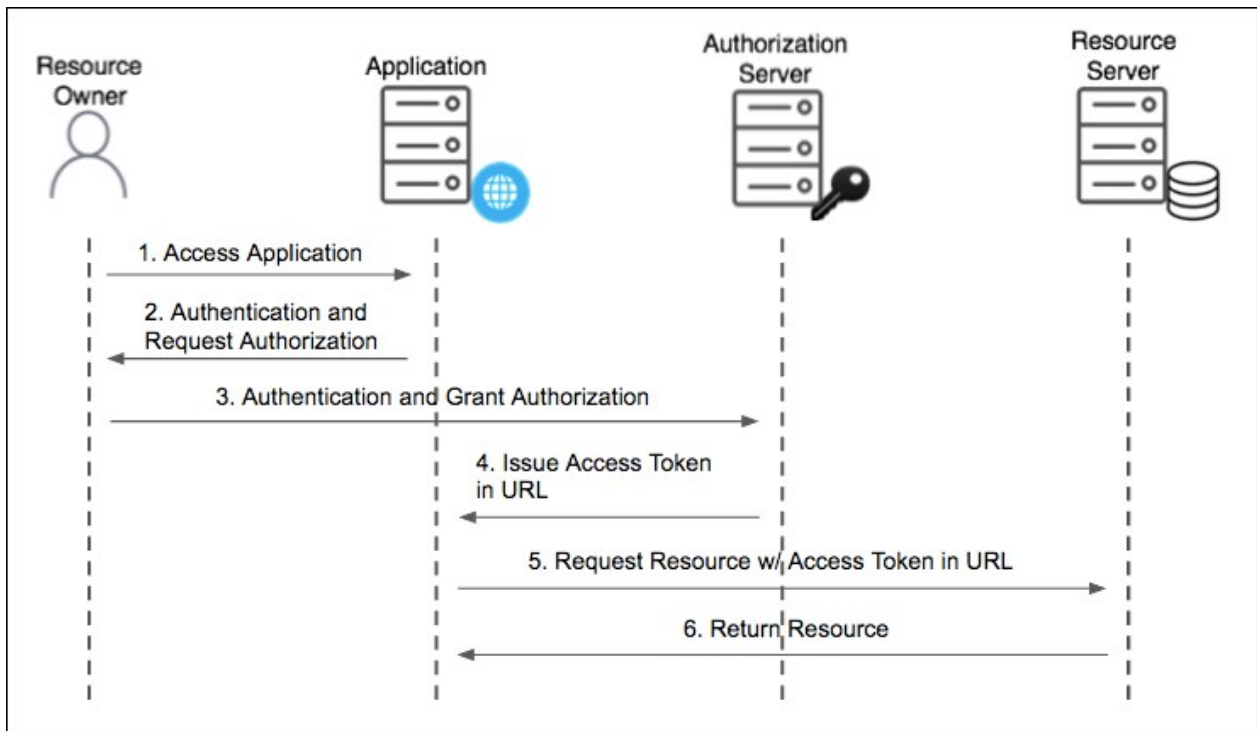
Implicit Grant Type Roles

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 Platform 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



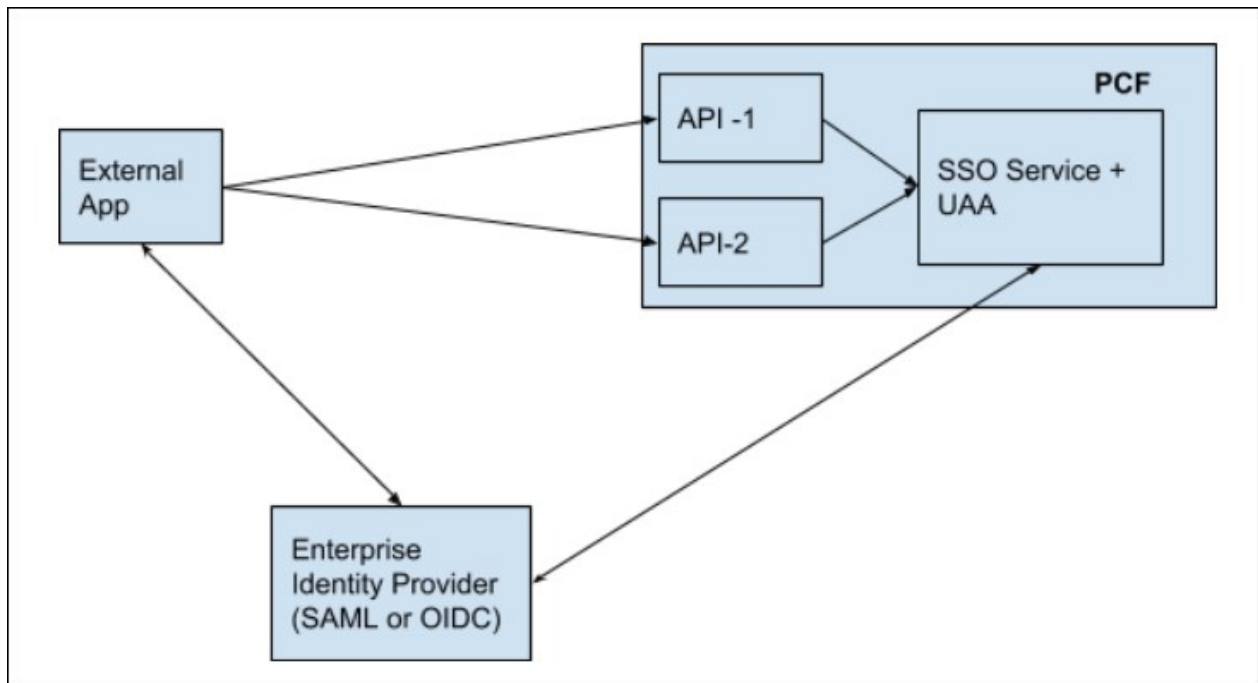
Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes common app architecture patterns for enterprise developers.

Externally Hosted Apps Call Pivotal Platform APIs

This section describes common architecture patterns for authenticating externally hosted apps that call Pivotal Platform APIs. In these patterns, externally hosted apps, secured either by Security Assertion Markup Language (SAML) or by OpenID Connect (OIDC) providers using JSON Web Tokens (JWTs), interact with Pivotal Single Sign-On to gain access to Pivotal Platform services.

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



In this diagram:

- You have an externally hosted app.
- The externally hosted app is secured by an enterprise SAML or OIDC identity provider, that is, users authenticate using the SAML or OIDC provider into the externally hosted app.

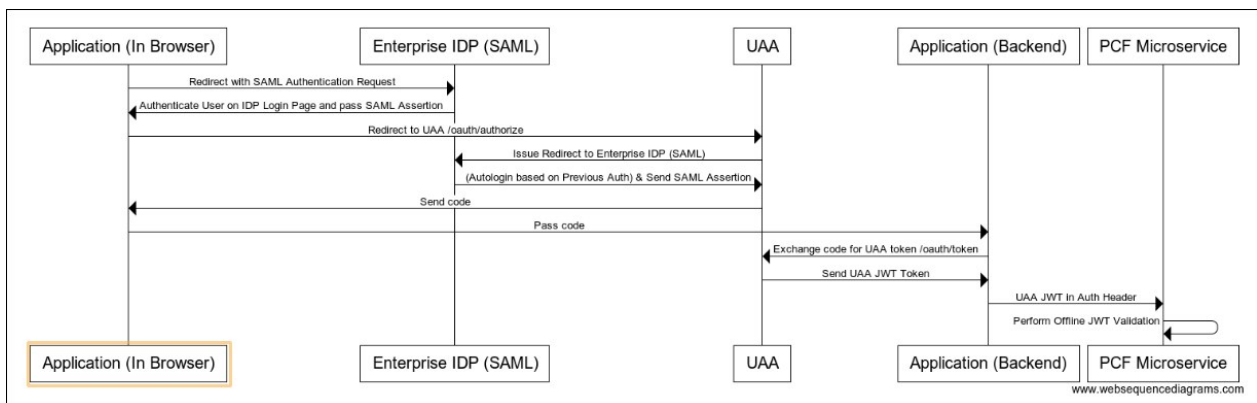
- The externally hosted app needs to invoke API-1 and API-2, which are Spring Boot microservices running on Pivotal Platform.
- API-1 and API-2 are protected by Single Sign-On using OAuth.

The following sections describe three authentication models that can be used for externally hosted apps using SAML or OIDC enterprise IDPs to call into Pivotal Platform 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.

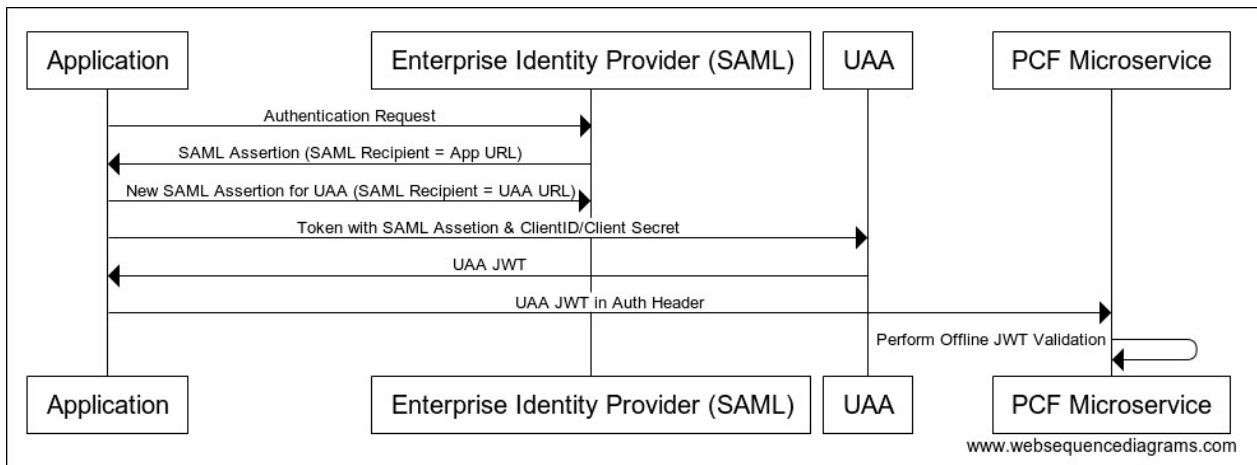


In this model:

- 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 Pivotal Platform 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 Pivotal Platform 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 Single Sign-On (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 Pivotal Platform 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>0TsVYrWJ4Yah1eM3p0e4DCLP3NlsgFoAZ6R/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
A1UECAwKQ2FsaWZvcmlkZXRhZARBNVBAAMCmRldi0yODEzNzYxHDAaBgkqhkiG9w0BCQEW
MBIGA1UECwwLU1NPUHJvdmlkZXRhZARBNVBAAMCmRldi0yODEzNzYxHDAaBgkqhkiG9w0BCQEW
DWluZm9Ab2t0YS5jb20WbHcNMTCwMTE4MTgwNTI5WWhcNMjcwMTE4MTgwNjI5WjCBkjELMAKGA1UE
BhMCVVMwEzARBgNVBAgMCkNhbm3JuaWEwEzFjAUBGNVBAcMDVNhbiBGcmFuY2l2Y28xDTALBgNV
BAoMBE9rdGEzFDASBgNVBASMC1NTT1Byb3ZpZGVyMRRwEQYDVQDDApkZXYtMjgxMzc2MRwwGyYJ
KoZIHvcNAQkBFglpbmZvQG9rdGEuY29tMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
0okZSvNaxO/QzpT92pxALWewO1j3F0DyFRjWz1x4u8AbPjJDizbr42pnm/dOxw5bij2CecvIvI3b
G/LNMh0NMB1uuMwRppIpNkU0mu/8b3ulszmGMSULRIAtCQFIKAd8VXApbmNlLsfzN5CnJzeDEZ29
3E/RGVr0WvSUKWYZaij57BfH2r2A44TZfRNPfUgtsvsVVQvtwDgKBo+rNqZIkQoMi0hdpX2Z522Z
16vpbDGu56kWR0fqyfoshKHPnHNvk/c0HkwcKIAm117DW95PnrTxjx7QQuLZibUFPD1sQE2S4Vxe
W6kxXhT8ttmML0OjirEqtD+98BcbqCc9SgYtzwIDAQABMA0GCSqGSIB3DQEBBQUAA4IBAQDKqs49
VBGPRTAWvGm+giBHT2uJd5JCefE6ap/OPp+ajfslXXH3yU0q6CiyKliVgS9j15MOVBDTou8vTtsK
wOTmdG1NHKJCjqpTe2h/+3uvCG2yv9D6rfDiQcO4KgeG+5hXnS2fGcFTuCuMODX7ivEYB9eeAqXkJG
4LFwxVhse8j0rwdkPESkdL7KdTbZK5rsM3tWihSsuccm4a6Zp6faFZzWhvd6ujBGilLtaVHP9jUG
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<

```

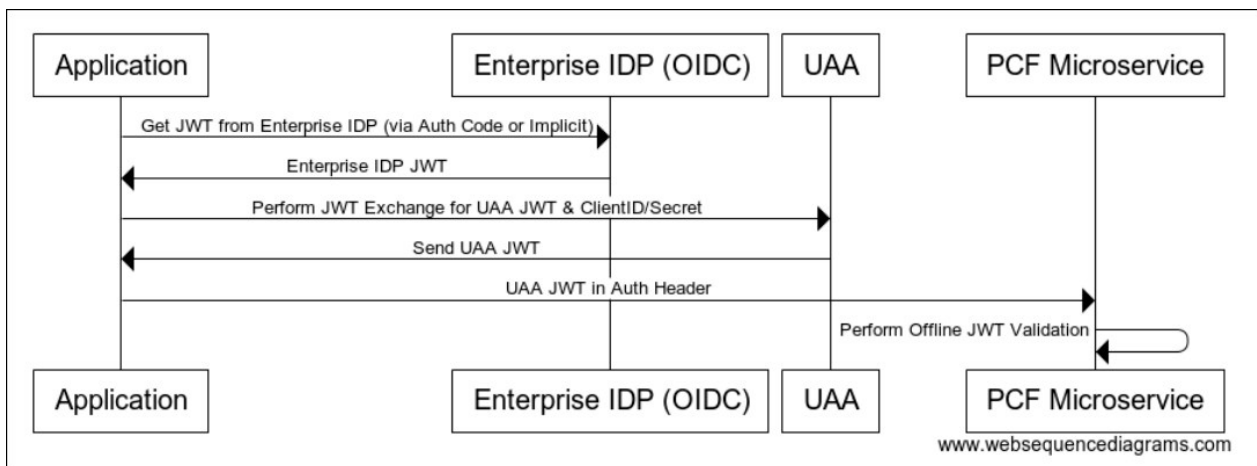
```

/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 externally hosted apps using OIDC.



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

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

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"
  ],
}

```

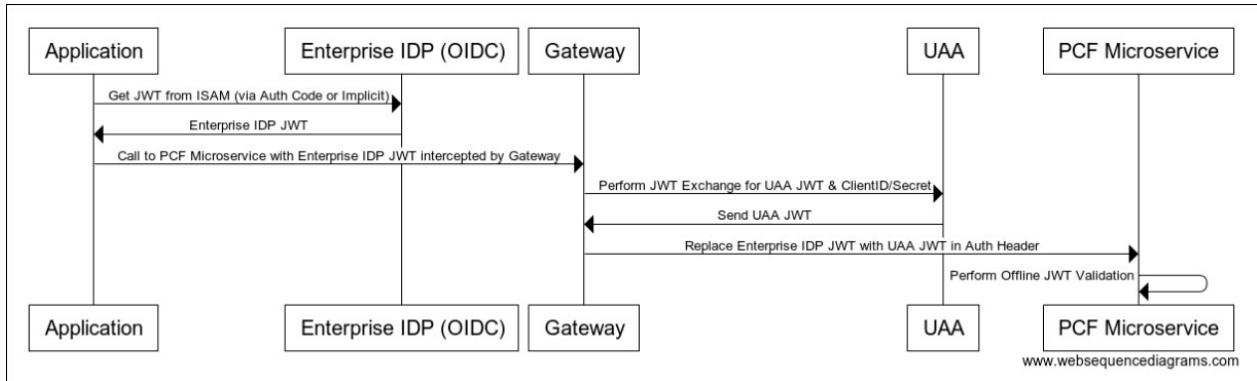
```

"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 Single Sign-On, 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 Single Sign-On service plan. Therefore, use the scope `clients.admin`.

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

3. Use UAAC to target the Single Sign-On 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 externally hosted 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 Single Sign- On Plan Administrator Dashboard:
 1. Add the enterprise SAML or OIDC (for JWT) provider.

For instructions, see [Configure 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 externally hosted app, or gateway, must make the following API call:
 1. Invoke the `/oauth/token` endpoint of the Single Sign- On service 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 externally hosted app
 - The SAML assertion or JWT token from the externally hosted 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On service plan in their space and bind it to an app.

Create Service Instances

1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN> as a Space Developer.
2. Navigate to the org that where the service plan is enabled.
3. Select **Marketplace** and select the **Single Sign-On** service.
4. Select the service plan you want and click **Select this plan**.
5. In the **Configure Instance** box, enter an **Instance Name**.
6. Under **Instance Configuration**, enter a name for your instance in **Instance Name**.
7. From the **Add to Space** drop-down, select a space for the instance. This space hosts your app.
8. From the **Bind to App** drop-down, select an app 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.
9. Click **Create** to create the service instance.

Access the SSO Developer Dashboard

1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN> as a Space Developer.
2. Navigate to the org and space where the service instance you want to manage is located.
3. Under **Services**, select the service instance you want to manage.
4. Click **Manage**.

Delete Service Instances

1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN> as a Space Developer.

2. Navigate to the org and space that contain the service instance you want to delete.
3. Click **Services** and then click the **Name** of the service you want to delete.
4. Click **Settings** and then click **Delete Service Instance**. On the pop-up, click **Delete Service Instance** again to confirm 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 apps 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: Pivotal Single Sign-On v1.11 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 developers configure their apps to use Pivotal Single Sign-On and use the SSO Admin Client to manage connections between SSO identity providers, apps, users, and other resources.

Overview

Single Sign-On enables users to log in through a single sign-on service and access other apps that are hosted or protected by the service. Eliminating the need for users to log in to individual apps improves security and productivity.

Register an App

Follow the instructions below to register an app for Single Sign-On.

Prerequisites for Registering Apps

Before you can register your app for Single Sign-On, you must create a service instance and choose your Single Sign-On app type as follows:

1. Create a service instance by doing the procedure in [Create Service Instances](#).
2. Determine your app's Single Sign-On app type. This informs which OAuth Grant Type you should use for your app. For more information, see [Determining Single Sign-On Application Type](#).



Note: If your app is a resource server, you do not need to register the app because it only consumes tokens and does not handle any token acquisition flow itself. A resource server hosts protected resources and accepts and responds to protected resource requests using access tokens, such as an API.

3. Choose a method to register your app with the Single Sign-On service:
 - If your app is hosted on Pivotal Application Service (PAS), follow one of the procedures in [Register a PAS App](#) below.

- ✦ If your app is not hosted on PAS, follow one of the procedures in [Register an Externally Hosted App](#) below.

Register a PAS App

To configure Single Sign-On for a PAS app, use one of the following methods:

- [Configure Single Sign-On Properties with JSON Bind Parameters](#)
- [Configure Single Sign-On Properties with the App Manifest](#)



Note: Apps that have environment variables set using the app manifest do not work with `cf bind-service`. If you have leftover environment variables previously set using the app manifest, you must remove them with `cf unset-env APP-NAME PROPERTY-NAME` before you perform a service bind.

Configure Single Sign-On Properties with JSON Bind Parameters

In Single Sign-On v1.7.0 and later, you can use a JSON file or string containing bind parameters to configure Single Sign-On properties. For more information about bind parameters, see [Bind Parameters](#) below.

To configure Single Sign-On properties with JSON bind parameters:

1. Run:

```
cf bind-service APP-NAME INSTANCE-NAME -c JSON
```

Where:

- ✦ `APP-NAME` is the name of your app
- ✦ `INSTANCE-NAME` is the name of the service instance you are binding
- ✦ `JSON` is the path to the JSON file or the full JSON string containing bind parameters

For example:

```
$ cf bind-service my-app my-instance -c example.json
Binding service my-instance to my-app in org my-org / space test as user@example.com...
OK

$ cat example.json
{
  "grant_types": ["authorization_code"],
  "scopes": ["openid", "todo.read", "todo.write"],
  "authorities": ["openid", "uaa.resource", "todo.read", "todo.write"],
  "redirect_uris": ["https://my-app.example.com/**", "http://my-app.example.com/path/to/app"],
  "auto_approved_scopes": ["openid", "todo.read"],
  "identity_providers": ["uaa", "ldap", "my-saml-provider"],
  "required_user_groups": ["openid", "todo.read"],
  "resources": {"todo.read": "Read to list", "todo.write": "Write to list"},
  "access_token_lifetime": 300,
  "refresh_token_lifetime": 86400,
```

```

"icon": "R0lGODlhAQABAAAAACH5BAEKAAEALAAAAABAAEAAICTAEAOw==",
"launch_url": "http://my-app.example.com",
"show_on_home_page": true
}

```

For more information about the `cf bind-service` command, see [bind-service](#) in the Cloud Foundry CLI Reference Guide.

You can also bind service instances with Apps Manager. For more information, see [Bind a Service Instance to an Application](#).

Bind Parameters

The table below provides descriptions and default values for bind parameters:

Property Name	Description	Default
<code>grant_types</code>	<p>The following grant types are supported by Single Sign-On:</p> <ul style="list-style-type: none"> <code>authorization_code</code> <code>password</code> <code>client_credentials</code> <code>implicit</code> <p>If you want to use more than one grant type, you can only use the <code>authorization_code</code> and <code>client_credentials</code> grant types at the same time. Otherwise, you can only use one grant type per app.</p>	<i>n/a</i>
<code>identity_providers</code>	<p>Allowed identity providers for the app through the Single Sign-On service plan. This is a comma-separated list of identity provider origin keys.</p> <p>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>redirect_uris</code>	<p>Comma-separated allowlist of redirection URIs allowed for the app. Each value must be a valid URI. Custom URIs are supported for mobile apps.</p>	Always includes the app route

<code>scopes</code>	Comma-separated list of scopes that belong to the app and are registered as client scopes with Single Sign-On. This value is ignored for client credential grant type apps.	<code>openid</code>
<code>auto_approved_scopes</code>	Comma-separated list of scopes that the app is automatically authorized when acting on behalf of users through Single Sign-On.	Defaults to existing <code>scopes/authorities</code>
<code>authorities</code>	Comma-separated list of authorities that belong to the app and are registered as client authorities with Single Sign-On. Privileged identity zone/plan admin scopes, such as <code>scim.read</code> and <code>idps.write</code> cannot be bootstrapped and must be assigned by zone/plan admins. This value is ignored for any grant type other than client credentials.	<code>uaa.resource</code>
<code>required_user_groups</code>	Comma-separated list of permissions a user must have to log in to the app. The user must have all of the listed permissions to access the app.	<i>n/a</i>
<code>access_token_lifetime</code>	Lifetime in seconds for the access token issued to the app by Single Sign-On.	<code>43200</code>
<code>refresh_token_lifetime</code>	Lifetime in seconds for the refresh token issued to the app by Single Sign-On.	<code>259200</code> (not used for client credentials)
<code>resources</code>	Resources for the app to use as scopes and authorities for Single Sign-On created during bind, if they do not already exist. 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 bind command. You cannot specify additional permissions in the same top level permission, such as <code>todo.admin</code> , in additional binds.	<i>n/a</i>
<code>icon</code>	App icon displayed next to the app name on the Account dashboard. It is displayed if <code>show_on_home_page</code> is set to <code>true</code> . Do not exceed 64kb.	<i>n/a</i>
<code>launch_url</code>	App launch URL used for the app on the Account dashboard if <code>show_on_home_page</code> is set to <code>true</code> .	The application route
<code>show_on_home_page</code>	If set to <code>true</code> , the app appears on the Account dashboard with the corresponding icon and launch URL.	<code>true</code>

Configure Single Sign-On Properties with the App Manifest

If you configure Single Sign-On properties with the app manifest, you do not need to manually bind your app to the service instance after you deploy your app. Single Sign-On reads its configuration properties from environment variables that are set in the apps that use it.



Note: These configurations are only applied at the time of the initial service bind. If you run `cf push` again, the app does not update the configurations. You can update the configurations through the SSO Developer Dashboard.

To register your app with the app manifest, do the following in your app manifest:

1. Specify the Single Sign-On service instance you want to bind your app. For an example, see the `services` section in the YAML sample below.
2. Configure the Single Sign-On configuration properties in your app manifest. For an example, see the properties in the `env` section in the YAML sample below.

The snippet below shows how to register your app with a Single Sign-On service instance and how to configure the `GRANT_TYPE` and `SSO_IDENTITY_PROVIDERS` Single Sign-On properties in your manifest.

```
---
applications:
  - name: my-example-app
    services:
      - my-sso-sample-instance
    env:
      GRANT_TYPE: implicit
      SSO_IDENTITY_PROVIDERS: uaa, sample-identity-provider
```

For examples of configuring Single Sign-On properties in an app manifest, see the `manifest.yml` files in the [identity-sample-apps](#) GitHub repository.

The table below provides descriptions and default values for Single Sign-On properties to include in the app manifest:

Property Name	Description	Default
<code>name</code>	Name of the app	None, but this value is required.

<code>GRANT_TYPE</code>	<p>The following grant types are supported by Single Sign- On:</p> <ul style="list-style-type: none"> <code>authorization_code</code> <code>password</code> <code>client_credentials</code> <code>implicit</code> <p>If you want to use more than one grant type, you can only use the <code>authorization_code</code> and <code>client_credentials</code> grant types at the same time. Otherwise, you can only use one grant type per app.</p>	<code>authorization_code</code> (Web application type)
<code>SSO_IDENTITY_PROVIDERS</code>	<p>Allowed identity providers for the app through the Single Sign- On 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> (Ops Manager internal user store)
<code>SSO_REDIRECT_URIS</code>	Comma-separated allowlist of redirection URIs for the app. Each value must be a valid URI. Custom URIs are supported for mobile apps.	(Always includes the app route)
<code>SSO_SCOPES</code>	<p>Comma-separated list of scopes that belong to the app and are registered as client scopes with Single Sign- On. This value is ignored for client credential grant type apps.</p> <p>VMware recommends including <code>openid</code>, which is not added to your user-provided list by default.</p>	<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 Single Sign- On.	(Defaults to existing scopes and authorities)

<code>SSO_AUTHORITIES</code>	Comma-separated list of authorities that belong to the app and are registered as client authorities with Single Sign- On. Privileged identity zone and plan admin scopes, such as <code>scim.read</code> , <code>idps.write</code> cannot be bootstrapped and must be assigned by zone and plan admins. 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 permissions a user must have to log in to the app. The user must have all of the listed permissions to access the app.	<i>n/a</i>
<code>SSO_ACCESS_TOKEN_LIFETIME</code>	Lifetime in seconds for the access token issued to the app by Single Sign- On.	<code>43200</code>
<code>SSO_REFRESH_TOKEN_LIFETIME</code>	Lifetime in seconds for the refresh token issued to the app by Single Sign- On.	<code>259200</code> (not used for client credentials)
<code>SSO_RESOURCES</code>	Resources that the app uses as scopes and authorities for Single Sign- On to be created during bootstrapping if they do not already exist. The input format can be referenced in the provided sample manifest. 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 app manifest. Currently you cannot specify additional permissions in the same top level permission, such as <code>todo.admin</code> , in additional app manifests.	<i>n/a</i>
<code>SSO_ICON</code>	App icon that is displayed next to the app name on the Account dashboard if Show on Homepage is enabled. Do not exceed 64kb.	<i>n/a</i>
<code>SSO_LAUNCH_URL</code>	App launch URL that is used for the app on the Account dashboard if Show on Homepage is enabled.	(App route)
<code>SSO_SHOW_ON_HOME_PAGE</code>	If set to <code>true</code> , the app appears on the Account dashboard with the corresponding icon and launch URL.	<code>true</code>

For more information and app manifest examples, see the [identity-sample-apps](#) GitHub repository.

Register an Externally Hosted App

To configure Single Sign- On for an app not hosted by PAS, use one of the following methods:

- [Register an Externally Hosted App Using Service Keys](#)
- [Register an Externally Hosted App Using the SSO Developer Dashboard](#)

Register an Externally Hosted App Using Service Keys

You can use a JSON file or string containing bind parameters to register an externally hosted app using service keys. For more information about bind parameters, see [Bind Parameters](#) above.

Service keys enable you to register apps for automation, such as scripts or pipelines.

To register an externally hosted app using service keys:

1. To create a service key, run:

```
cf create-service-key INSTANCE-NAME SERVICE-KEY -c JSON
```

Where:

- ✦ `INSTANCE-NAME` is the name of your service instance.
- ✦ `SERVICE-KEY` is the name you want for your service key.
- ✦ `JSON` is the path to the JSON file or the full JSON string containing bind parameters.

You must include the `name` parameter in your JSON file or string. The value of `name` is the user-defined name of your externally hosted app.

For example:

```
$ cf create-service-key my-instance my-service-key -c example.json
Creating service key my-service-key for service instance my-instance as user@example.com...
OK

$ cat example.json
{
  "name": "my-app",
  "grant_types": ["authorization_code"],
  "scopes": ["openid", "todo.read", "todo.write"],
  "authorities": ["openid", "uaa.resource", "todo.read", "todo.write"],
  "redirect_uris": ["https://my-app.example.com/**", "http://my-app.example.com/path/to/app"],
  "auto_approved_scopes": ["openid", "todo.read"],
  "identity_providers": ["uaa", "ldap", "saml"],
  "required_user_groups": ["openid", "todo.read"],
  "resources": {"todo.read": "Read to list", "todo.write": "Write to list"},
  "access_token_lifetime": 300,
  "refresh_token_lifetime": 86400,
  "icon": "R0lGODlhAQABAAAAACH5BAEKAAEALAAAAABAAEAAAICTAEAOw==",
  "launch_url": "http://google.com",
  "show_on_home_page": false
}
```

For more information about the `cf create-service-key` command, see [create-service-key](#) in the Cloud Foundry CLI Reference Guide.

2. To obtain the service key credentials, run:

```
cf service-key INSTANCE-NAME SERVICE-KEY
```

Where:

- ✦ `INSTANCE-NAME` is the name of your service instance.
- ✦ `SERVICE-KEY` is the name of the service key you created in step 1.

For more information about how to use these values, see [Integrate Single Sign-On with Your App](#).

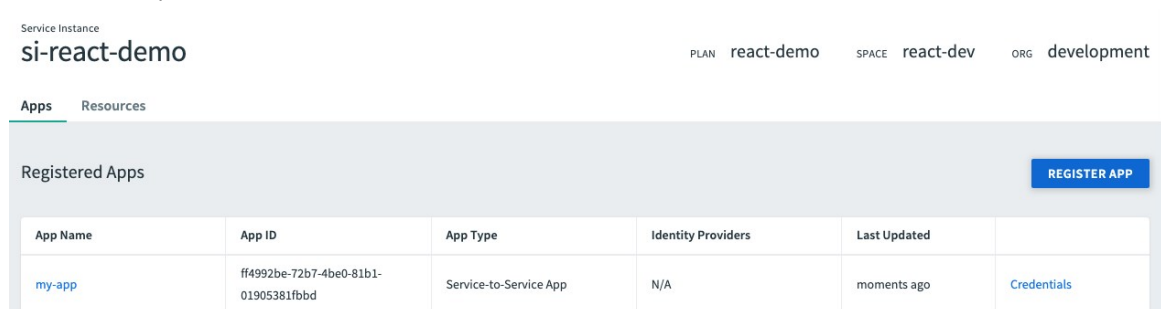


Note: `client_credentials` grant types, administrative scopes, and scopes in other spaces are not available. These must be manually added by a plan admin or system operator through the SSO Operator Dashboard or UAA CLI.

Register an Externally Hosted App Using the SSO Developer Dashboard

To register an externally hosted app using the SSO Developer Dashboard:

1. Determine the Single Sign-On app type for your app. For more information, see [Determining Single Sign-On Application Type](#).
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 Single Sign-On service instance. This launches the SSO Developer Dashboard.




[View a larger version of this image.](#)

5. Click **Register App**.
6. Enter an **App Name**.
7. Choose an app type under **Select an App Type**.
8. Under **Select Identity Providers**, select an identity provider 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 app types except `Service-to-Service App`.

Select Identity Providers

Identity Provider	Type	Origin Key
<input checked="" type="checkbox"/> Internal User Store	 UAA	uaa

9. If your App Type is **Web App** or **Single-Page JavaScript App**, under **App Settings** enter an allowlist of URIs beneath **Redirect URIs Whitelist**. The redirect query parameter specified on the OAuth request must match the URIs specified in this list. Otherwise, Single Sign-On rejects the request.

App Settings

Provide a whitelist of URIs that are valid during login and logout redirects

Redirect URI Whitelist

Provide a comma-separated list of URIs. Each URI must start with a URI scheme. Example URI Schemes: `https://`, `custom-scheme://`

10. Under **Authorization**, select the **System Permissions** that the app can request on the user's behalf. If this app is only 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 that the API enforces and the scopes that the app requests.

Scope	Description
<code>open</code>	Provides access to make OpenID Connect request. The default for Web, Native, and Single-Page JavaScript Apps
<code>user_attributes</code>	Provides access to custom attributes from an external identity provider
<code>roles</code>	Provides access to external groups from an identity provider
<code>uaa.resource</code>	Provides access to the <code>check_token</code> endpoint for service-to-service flows. The default for Service-to-Service Apps.



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 app type is a **Web App**, **Native App**, or **Single-Page JavaScript App**. If the app type is a **Service-to-Service App**, you can only select resources defined within the space.

11. (Optional) Under **Auto-Approved Scopes**, select any scopes that Single Sign-On 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 externally hosted apps.

12. (Optional) Under **Authentication**, select the permissions that users must have to log in to the app. The user must have all of the selected permissions to access the app.



Note: The **Authentication** settings do not apply to the **Service-to-Service App** app type.

13. (Optional) Under **Token Validity**, change the token expiration times. The default is set in the configuration for the Single Sign- On service plan. For more information about tokens, see [Configure a Token Policy](#).
14. (Optional) If your App Type is a **Web App** or **Single-Page JavaScript App**, you can enable **Show on Homepage** to display the app on the UAA or Account home page. If you want an app to display on the home page, you must enter an **App Launch URL** or upload an app icon.
15. In **App Launch URL**, enter the address you want for your app.
16. Upload an app icon for your app.

Show on Homepage

☒ When a user logs directly into the login page, the user can see a list of apps to access

App Launch Url (Optional)

App Launch URL must start with http:// or https://

App Icon (Optional)

File size limit is 1MB and must be a JPEG, GIF, or PNG

UPLOAD APP ICON

REMOVE

17. Click **Register App**.
18. View and download the **App ID** and **App Secret**. Record these credentials to use in other Single Sign- On procedures.



Note: You can only view the **App Secret** when an app is first registered or the secret is regenerated.

19. (Optional) To view examples of how to integrate Single Sign- On into your Spring Boot apps, click **Download Sample Apps**.

Easily test these settings with a ready-to-go Spring Boot app.

DOWNLOAD SAMPLE APPS

Manage App Configurations

The SSO Developer Dashboard enables app developers to view the app configurations and resources available within their space.

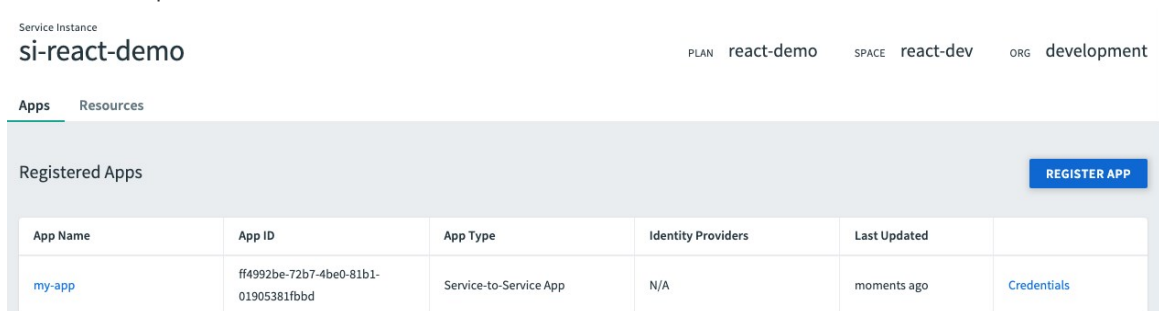
Prerequisite

Before you access the dashboard, you must create a service instance for your space. See [Create Service Instances](#).

Procedure

To manage your app configurations through the SSO Developer 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 Single Sign- On service instance to launch the SSO Developer Dashboard.



[View a larger version of this image.](#)

4. Click your app.
5. Under **Select Identity Providers**, select an identity provider for your app. Internal User Store is the default.



Note: When binding a PAS 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 app types except [Service-to-Service App](#).

Select Identity Providers

Identity Provider	Type	Origin Key
<input checked="" type="checkbox"/> Internal User Store	UAA	uaa

6. If your App Type is [Web App](#) or [Single-Page JavaScript App](#), under **App Settings** enter an allowlist of URIs beneath **Redirect URIs Whitelist**. The redirect query parameter specified on the OAuth request must match the URIs specified in this list. Otherwise, Single Sign- On rejects the request.

App Settings

Provide a whitelist of URIs that are valid during login and logout redirects

Redirect URI Whitelist

Provide a comma-separated list of URIs. Each URI must start with a URI scheme. Example URI Schemes: `https://`, `custom-scheme://`

- Under **Authorization**, select the **System Permissions** that the app can request on the user's behalf. If this app is only 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 that the API enforces and the scopes that the app requests.

Scope	Description
<code>open</code>	Provides access to make OpenID Connect request. The default for Web, Native, and Single-Page JavaScript Apps
<code>user_attributes</code>	Provides access to custom attributes from an external identity provider
<code>roles</code>	Provides access to external groups from an identity provider
<code>uaa.resource</code>	Provides access to the <code>check_token</code> endpoint for service-to-service flows. The default for Service-to-Service Apps.



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

- (Optional) Under **Auto-Approved Scopes**, select any scopes that Single Sign-On 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 externally hosted apps.
- (Optional) Under **Token Validity**, change the token expiration times. The default is set in the configuration for the Single Sign-On service plan. For more information about tokens, see [Configure a Token Policy](#).
- (Optional) If your App Type is a `Web App` or `Single-Page JavaScript App`, you can enable **Show on Homepage** to display the app on the UAA or Account home page.



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

- In **App Launch URL**, enter the address you want for your app.
- Upload an app icon for your app.

Show on Homepage

☒ When a user logs directly into the login page, the user can see a list of apps to access

App Launch Url (Optional)

App Launch URL must start with http:// or https://

App Icon (Optional)

File size limit is 1MB and must be a JPEG, GIF, or PNG


13. Click **Update App**.

Credentials of Existing App Configurations

You can view the credentials and endpoints required for app integration for an existing app on the **Credentials** page.

To view credentials of an existing app:





1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click the **Single Sign- On** service.
4. Click **Manage** next to your Single Sign- On service instance to launch the SSO Developer Dashboard.
5. Under **Apps**, click **Credentials** near the name of your app.



Note: You can only view the **App Secret** when an app is first registered or the secret is regenerated.

Service-to-Service App

my-app

App ID Unique identifier for the application	<input type="text"/>	
App Secret Authenticates the application	<input type="text"/>	
The App Secret can only be displayed once. Regenerate app secret		
SSO Service URL Auth domain for single sign-on	<input type="text"/>	
OAuth Token URL Client retrieves token from this endpoint	<input type="text"/>	
Token Verification Keys An endpoint which returns JWT verification keys	<input type="text"/>	

[View a larger version of this image.](#)

Regenerate an App Secret



Warning: You must not regenerate an app secret for PAS bound apps, because `VCAP_SERVICES` does not update with the new `client_secret`. To regenerate a client for a PAS bound app, you must delete and re-create the service binding from your app to Single Sign- On.

To regenerate an app secret of an existing app:

1. Log in to Apps Manager as a Space Developer.
2. Select the space where your service instance is located.
3. Under **Services**, click the **Single Sign- On** service.
4. Click **Manage** next to your Single Sign- On service instance to launch the SSO Developer Dashboard.
5. Under **Apps**, click the **Credentials** near the name of your app.
6. Click **Regenerate App secret** under the **App Secret** field.
7. On the dialog box, click **Regenerate App Secret** to confirm that you want to regenerate an app's secret value.
8. View and download the **App ID** and regenerated **App Secret**.
9. Record these credentials to use in other Single Sign- On procedures.



Note: Regenerating an **App Secret** requires you to update the secret at the client.

Create an 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 admin to follow 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 Single Sign- On service instance to launch the SSO Developer Dashboard.
5. Click **Register App**.
6. Enter an **App Name**.
7. Under **Select an App Type**, select **Service-to-Service App**.
8. Under **Authorization**, select 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

9. Click **Register App**.
10. View and download the **App ID** and **App Secret**.
11. Record these credentials to use in other Single Sign- On procedures.



Note: You can only view the **App Secret** when an app is first registered or the secret is regenerated.

Unregister App from Single Sign- On

To unregister an app from Single Sign- On:

- If you configured a PAS app in [Set Up Apps to Use Single Sign- On](#) above, see [Unregister a PAS App](#) below.
- If you registered an externally hosted app using service keys in [Register an Externally Hosted App Using Service Keys](#) above, see [Unregister an Externally Hosted App Using Service Keys](#) below.
- If you registered an externally hosted app using the SSO Developer Dashboard in [Register an Externally Hosted App Using the SSO Developer Dashboard](#) above, see [Unregister an Externally Hosted App Using the Service Instance Developer Dashboard](#) below.

Unregister a PAS App

To unregister a PAS app:

1. Run:

```
cf unbind-service APP-NAME INSTANCE-NAME
```

Where:

- ♦ `APP-NAME` is the name of your app
- ♦ `INSTANCE-NAME` is the name of the service instance you are binding

For more information about the `cf unbind-service` command, see [unbind-service](#) in the Cloud Foundry CLI Reference Guide.



Warning: When you unbind the app, all SSO configuration for the app is permanently deleted.



Note: If you delete a PAS app, the app is automatically unregistered from Single Sign-On.

Unregister an Externally Hosted App Using Service Keys

To unregister an externally hosted app using service keys:

1. Run:

```
cf delete-service-key INSTANCE-NAME SERVICE-KEY
```

Where:

- ✦ `INSTANCE-NAME` is the name of your service instance
- ✦ `SERVICE-KEY` is the name you want for your service key

For more information about the `cf delete-service-key` command, see [delete-service-key](#) in the Cloud Foundry CLI Reference Guide.

Unregister an Externally Hosted App Using SSO Developer Dashboard

To unregister an externally hosted app using the SSO Developer Dashboard:

1. To navigate to the SSO Developer Dashboard, follow the procedure in [Access the SSO Developer Dashboard](#).
2. Click your app.
3. Click **Unregister App** at the bottom of the page.
4. When prompted, click **Unregister App** to confirm that you want to unregister the app and its configurations.



Note: Deleting an externally hosted app removes the app and its configurations from the SSO Developer 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On 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 app manifest or defined by the Space Developer in the SSO Developer Dashboard.

To bootstrap resources from the manifest, follow the instructions in the [Single Sign-On Sample Applications repository](#).

To create resources in the SSO Developer Dashboard, do the following:

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 Single Sign-On service instance to launch the SSO Developer Dashboard.

Resources			CREATE RESOURCE
Resource	Permissions	Last Updated	
todo	todo.read, todo.write	1 week ago	
test-new-res	test-new-res.test	4 days ago	
some-resource	some-resource.some-permission, some-resource.a.b	1 month ago	
taxform	taxform.write, taxform.read, taxform.clone	1 month ago	
rabbitmq	rabbitmq.read:/*	1 week ago	

[View a larger version of this image.](#)

4. Click the **Resources** tab.
5. Click **Create 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 **Create Resource**.
9. An administrator can map existing groups to the created resource. For more information, see [Create or Edit Resource Permissions Mapping](#).



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 Single Sign- On service instance to launch the SSO Developer Dashboard.

Resources CREATE RESOURCE		
Resource	Permissions	Last Updated
todo	todo.read, todo.write	1 week ago
test-new-res	test-new-res.test	4 days ago
some-resource	some-resource.some-permission, some-resource.a.b	1 month ago
taxform	taxform.write, taxform.read, taxform.clone	1 month ago
rabbitmq	rabbitmq.read:/*	1 week ago

[View a larger version of this image.](#)

3. Click the **Resources** tab.
4. Click the resource to delete.
5. Click **Delete Resource** 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.

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 a 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](#)

Integrating Pivotal Single Sign- On with Your App



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign- On with Java and non-Java apps.

Integrate Single Sign- On with an App

Because Single Sign- On is based on the OAuth protocol, any app that uses Single Sign- On must be OAuth-aware.

Java Apps

If you are using Java, see the [Single Sign- On Service Sample Applications](#) repository. These are sample [Spring Boot](#) apps that demonstrate how to use Single Sign- On Service libraries to configure the app for OAuth.

For Spring Boot 1.5, use [spring-cloud-sso-connector](#) and see the [spring-boot-1.5](#) branch of the [Single Sign- On Service Sample Applications](#) repository for examples.

For Spring Boot 2.1, use [java-cfenv-boot-pivotal-sso](#) and see the [spring-boot-2.1](#) branch of the [Single Sign- On Service Sample Applications](#) repository for examples.

After binding the app to a Single Sign- On service instance, you must restart the app for the new Single Sign- On 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 you bind the app to a Single Sign- On service instance. You can view this information on the **Next Steps** page of the SSO Developer 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` uses the following pattern:

```
https://AUTH-DOMAIN.uaa.SYSTEM-DOMAIN/oauth/token
```

Where `AUTH-DOMAIN` is the **Auth Domain** you entered in [Create or Edit Service Plans](#).

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

Login Hints

When you make an authorization code, password or implicit grant request, a login hint can be provided so that the end user is automatically redirected to the appropriate identity provider.

An encoded JSON string containing `origin_key` tied to the origin key of an identity provider can be provided as a login hint using `login_hint` in a query parameter. For information about login hints, see the [Authorization Code Grant](#) in the UAA API documentation.

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

Active Directory Federation Services Integration Guide Overview



Warning: Pivotal Single Sign-On v1.11 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 and Pivotal Single Sign-On as the service provider.

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

Prerequisites

To integrate AD FS with Pivotal Platform, you must have the following:

- An AD FS 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 Single Sign-On

Complete both steps below to integrate your deployment with AD FS and Single Sign-On.

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

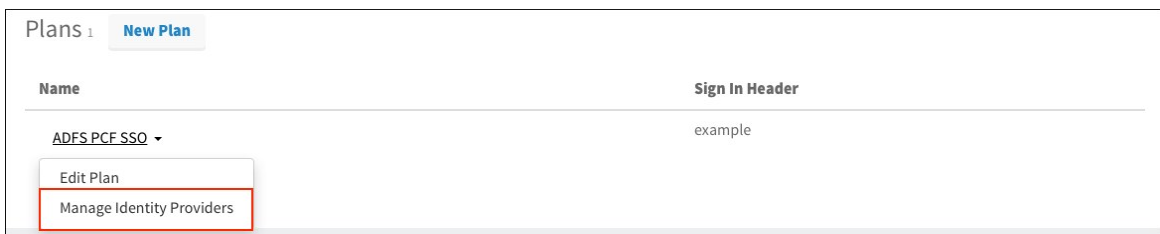


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

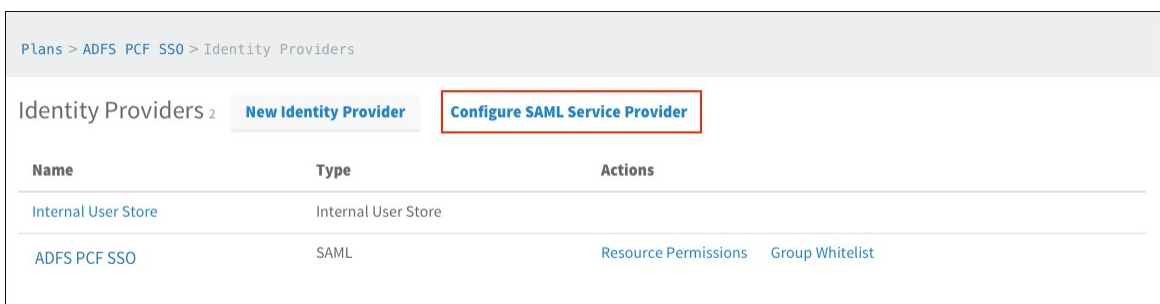
This topic describes how to set up Active Directory Federation Services (AD FS) as your identity provider by configuring SAML integration.

Set Up SAML with the SSO Operator Dashboard

1. Log in to the SSO Operator Dashboard at <https://p-identity.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 single sign-on private key signature and identity provider validation.

Configure SAML Service Provider

[Download Metadata](#)

- ☒ Perform signed authentication requests
- ☐ Require signed assertions

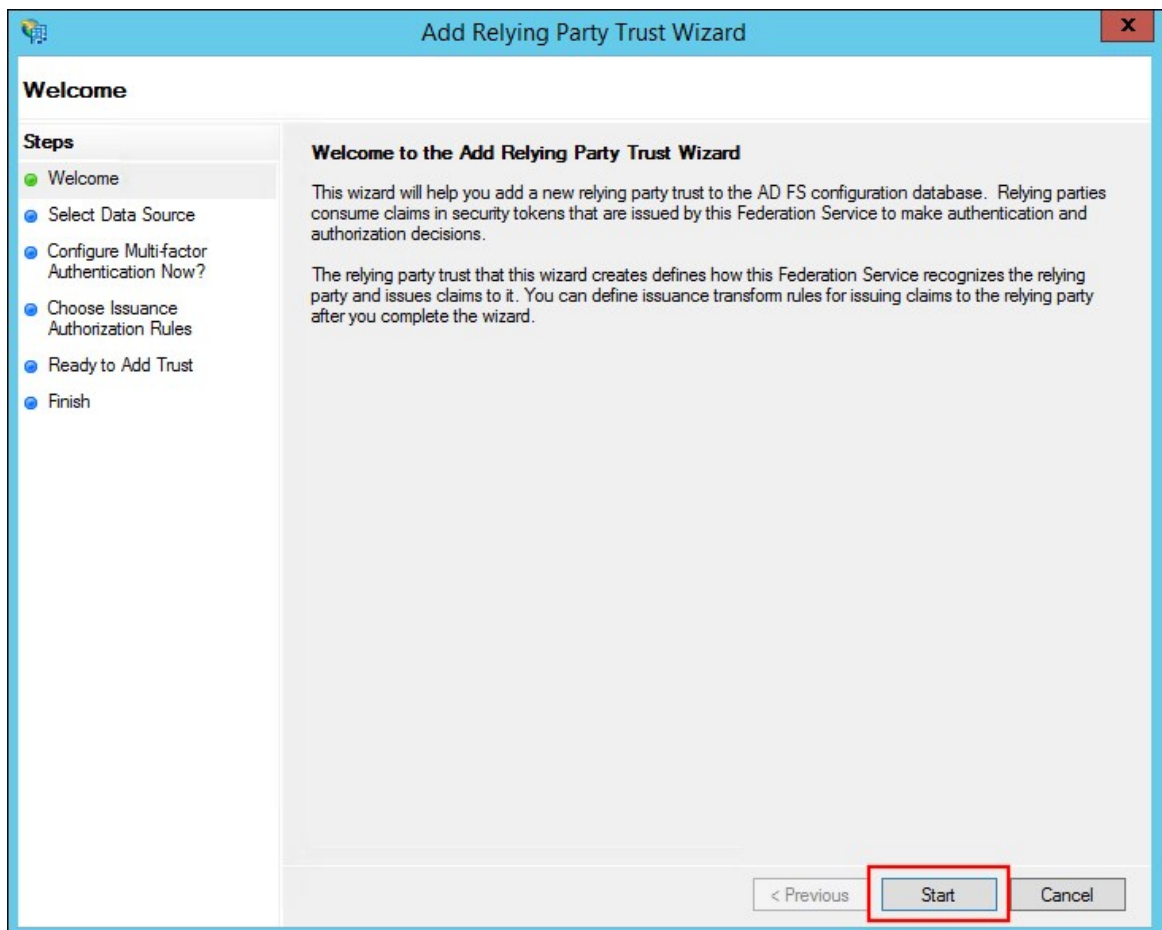
Cancel

Save

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

Set Up SAML in AD FS

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



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

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar is blue with a close button. The main window has a light blue header with the title 'Add Relying Party Trust Wizard'. Below the header is a sidebar on the left titled 'Select Data Source' with a 'Steps' section. The steps are: Welcome, Select Data Source (highlighted), Specify Display Name, Configure Multi-factor Authentication Now?, Choose Issuance Authorization Rules, Ready to Add Trust, and Finish. The main area is titled 'Select an option that this wizard will use to obtain data about this relying party:'. There are three radio button options. The first is 'Import data about the relying party published online or on a local network' with a description and a text box for 'Federation metadata address (host name or URL)'. The second is 'Import data about the relying party from a file' (selected and highlighted with a red box) with a description and a text box for 'Federation metadata file location' containing 'C:\Users\Administrator\Downloads\spring_saml_metadata.xml' and a 'Browse...' button. The third is 'Enter data about the relying party manually' with a description. At the bottom are buttons for '< Previous', 'Next >', and 'Cancel'.

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**.

The screenshot shows the 'Add Relying Party Trust Wizard' window. The title bar is blue with the text 'Add Relying Party Trust Wizard' and a close button. The main area is titled 'Specify Display Name'. On the left, a 'Steps' pane lists the following steps: 'Welcome' (green dot), 'Select Data Source' (green dot), 'Specify Display Name' (green dot, currently selected), 'Configure Multi-factor Authentication Now?' (blue dot), 'Choose Issuance Authorization Rules' (blue dot), 'Ready to Add Trust' (blue dot), and 'Finish' (blue dot). The main content area has a header 'Enter the display name and any optional notes for this relying party.' Below this is a 'Display name:' label followed by a text box containing 'ADFS PCF SSO'. Below the text box is a 'Notes:' label followed by a large text area. At the bottom right, there are three buttons: '< Previous', 'Next >', and 'Cancel'.

Add Relying Party Trust Wizard

Specify Display Name

Steps

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

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

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

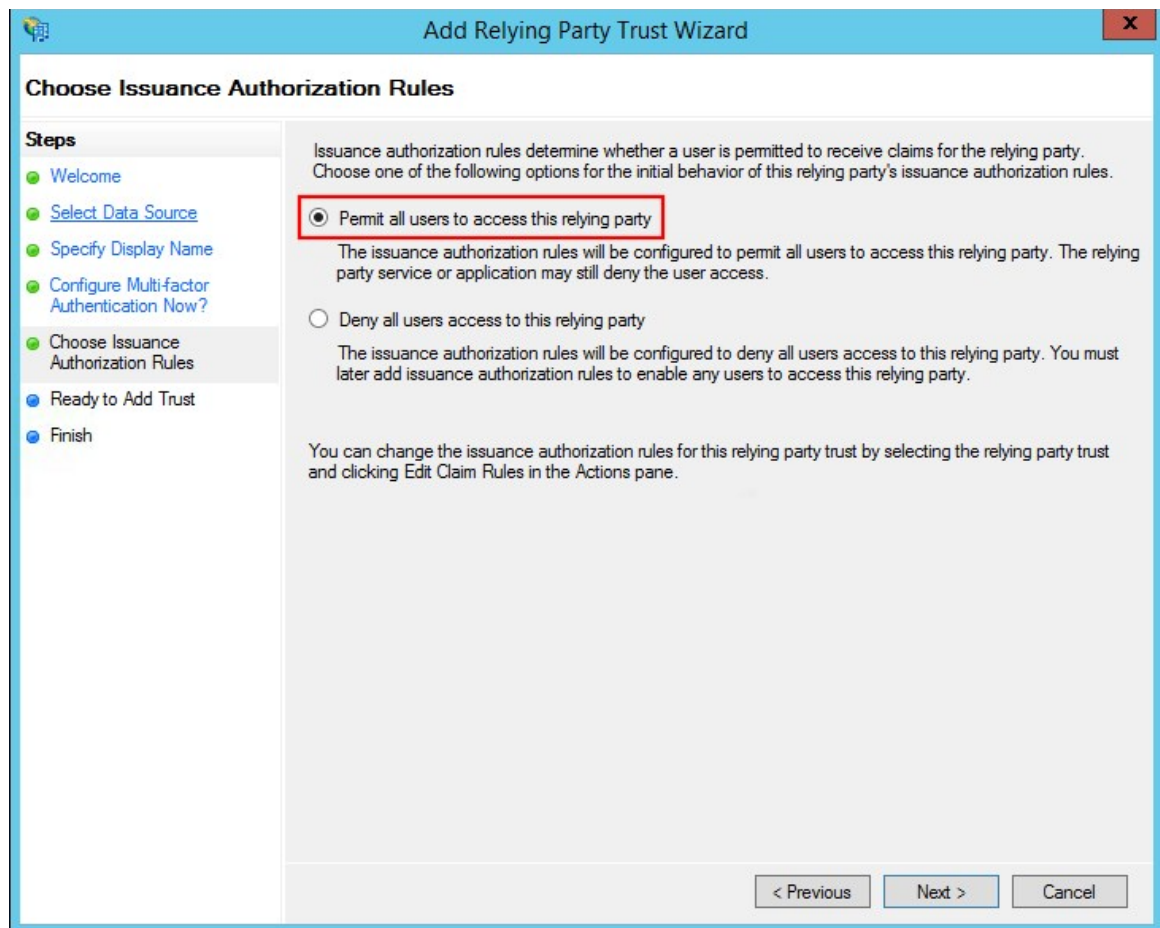
☒ I do not want to configure multi-factor authentication settings for this relying party trust at this time.

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

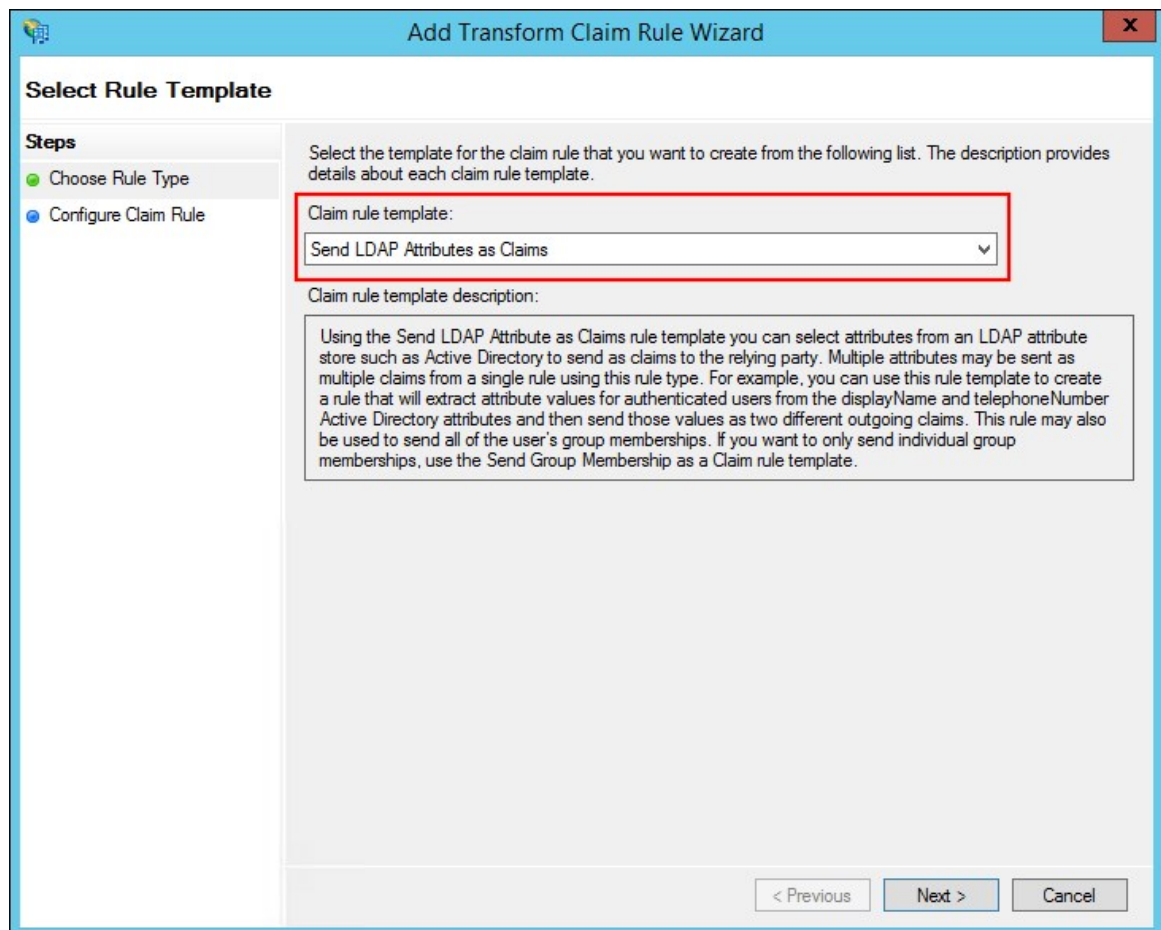
You can also configure multi-factor authentication settings for this relying party trust by navigating to the Authentication Policies node. For more information, see [Configuring Authentication Policies](#).

< 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**.



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

1. Click **Add Rule**. 2. Select **Transform an Incoming Claim** 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. The main area is titled "Select Rule Template" and contains the 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 "Transform an Incoming Claim" selected. A red rectangle highlights this dropdown. Below the dropdown is a text box titled "Claim rule template description:" containing the following text: "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." At the bottom right are three buttons: "< Previous", "Next >", and "Cancel".

3. Enter a **Claim rule name**. 4. Select **E-Mail Address** for **Incoming claim type**. 5. Select **Name ID** for **Outgoing claim type** 6. Select **Email** for **Outgoing name ID format**. 6. 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

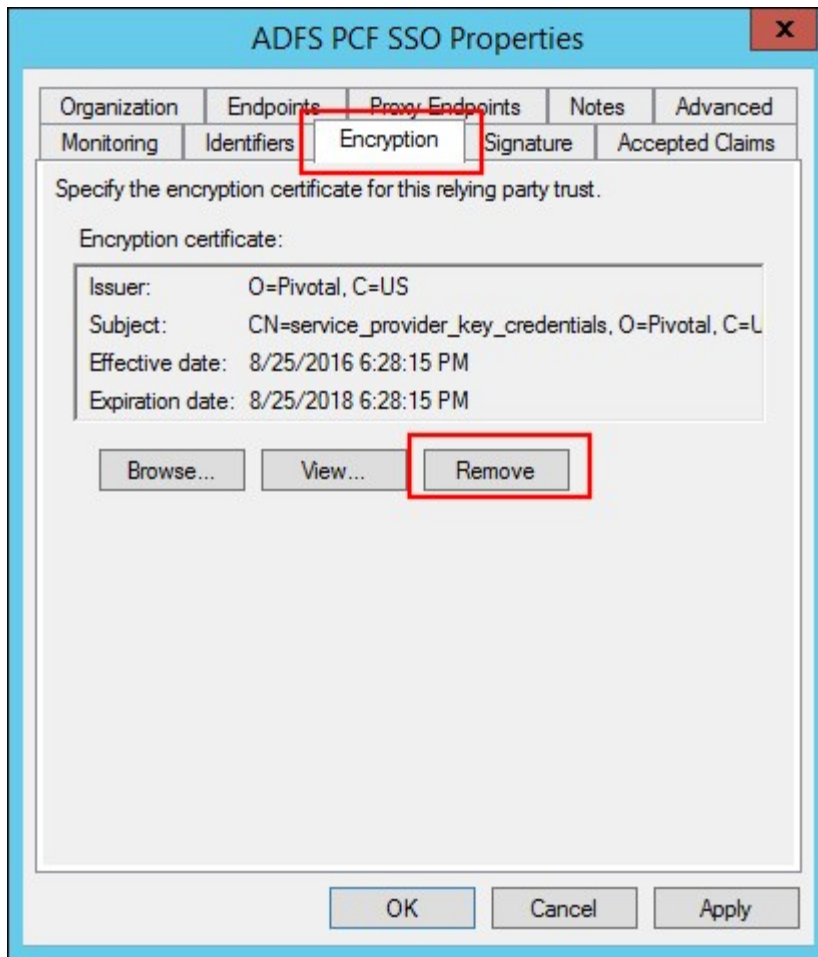
Outgoing claim type: Name ID

Outgoing name ID format: Email

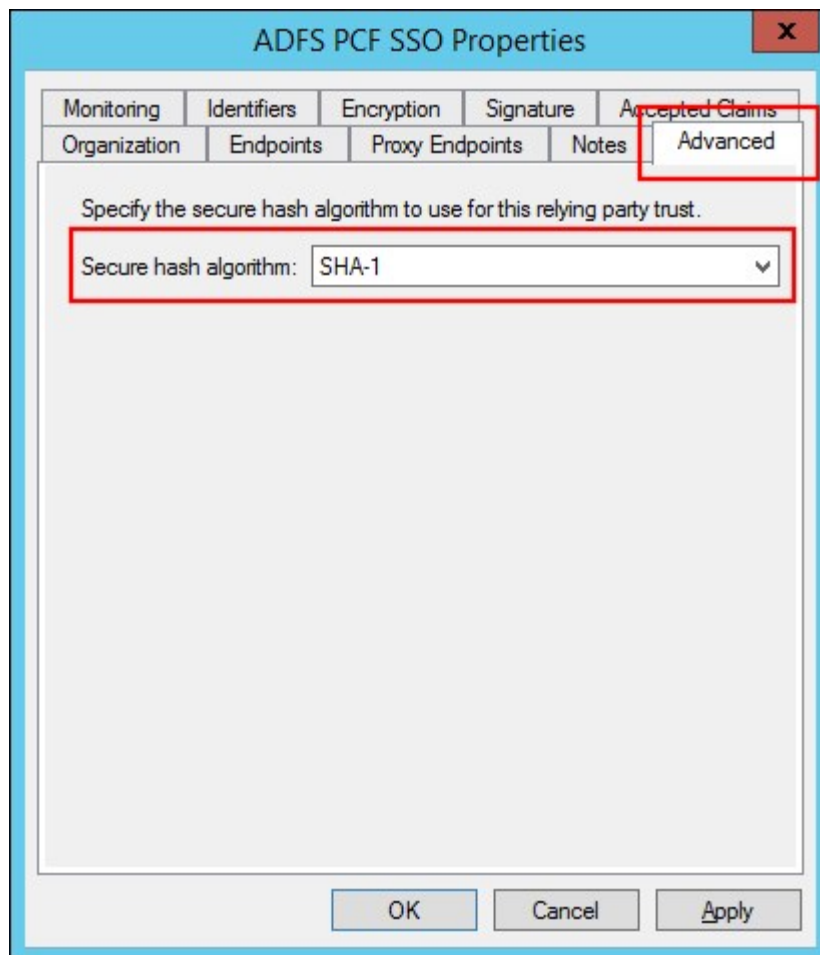
☒ Pass through all claim values
☐ Replace an incoming claim value with a different outgoing claim value
 Incoming claim value:
 Outgoing claim value: Browse...
☐ Replace incoming e-mail suffix claims with a new e-mail suffix
 New e-mail suffix:
 Example: fabrikam.com

< Previous Finish Cancel

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

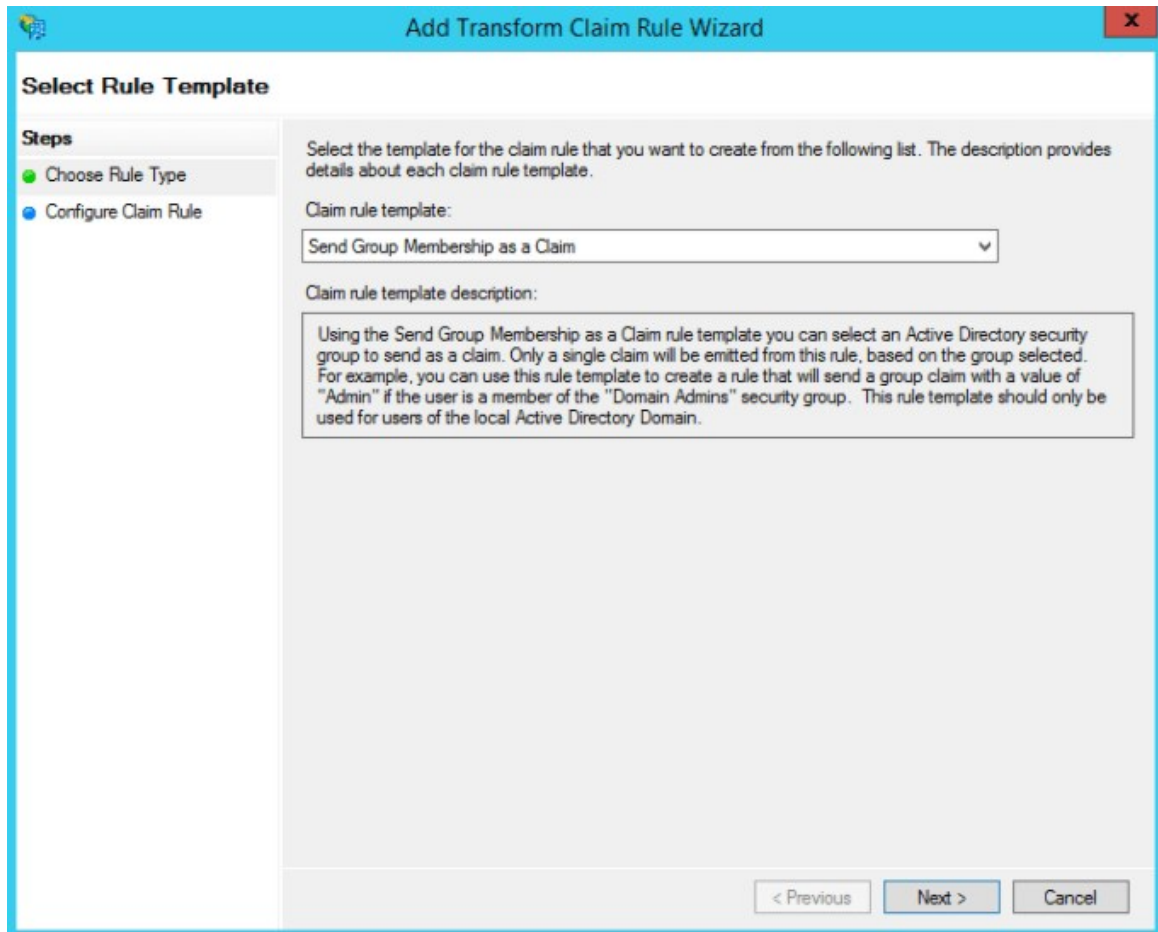
16. (Optional) If you are using a self-signed certificate, add it to the AD FS trust store. Obtain the Ops Manager certificate from https://OPS_MANAGER_IP/api/v0/security/root_ca_certificate and add this CA certificate to the AD FS trust store, so AD FS can trust the "Service Provider Key Certificate"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

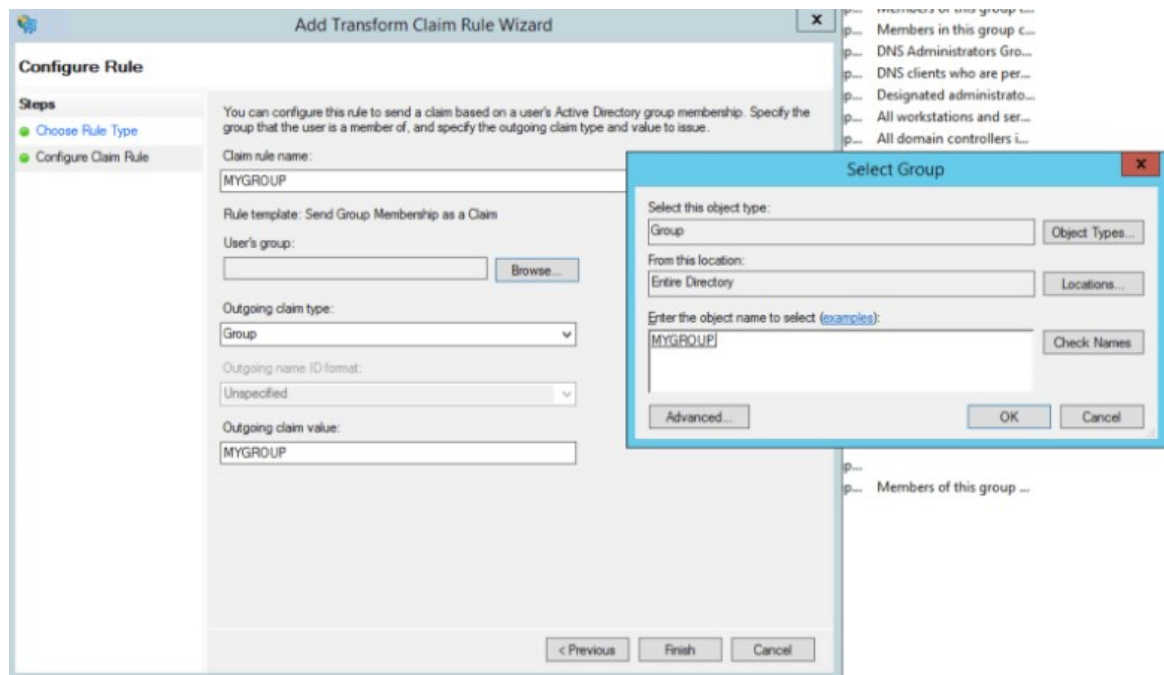
1. Right-click your **Relying Party Trust** and select **Edit Claim Rules...**



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



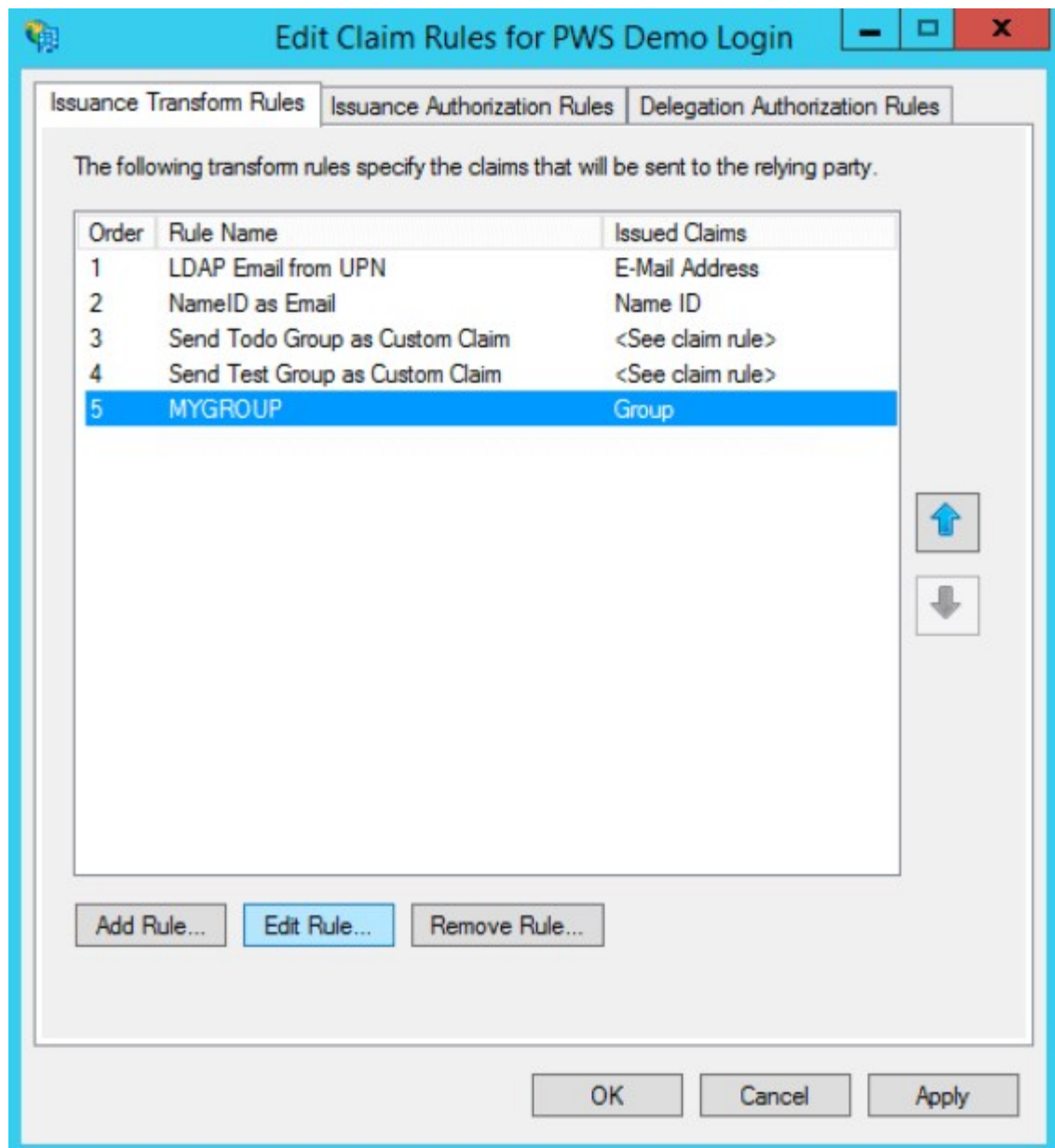
4. Enter the **Claim rule name**.
5. Click **Browse** to select your **User' s group**.
6. Select **Group** as your **Outgoing claim type**.
7. Set your **Outgoing claim value** to match your group' s name.
8. Click **Finish**.



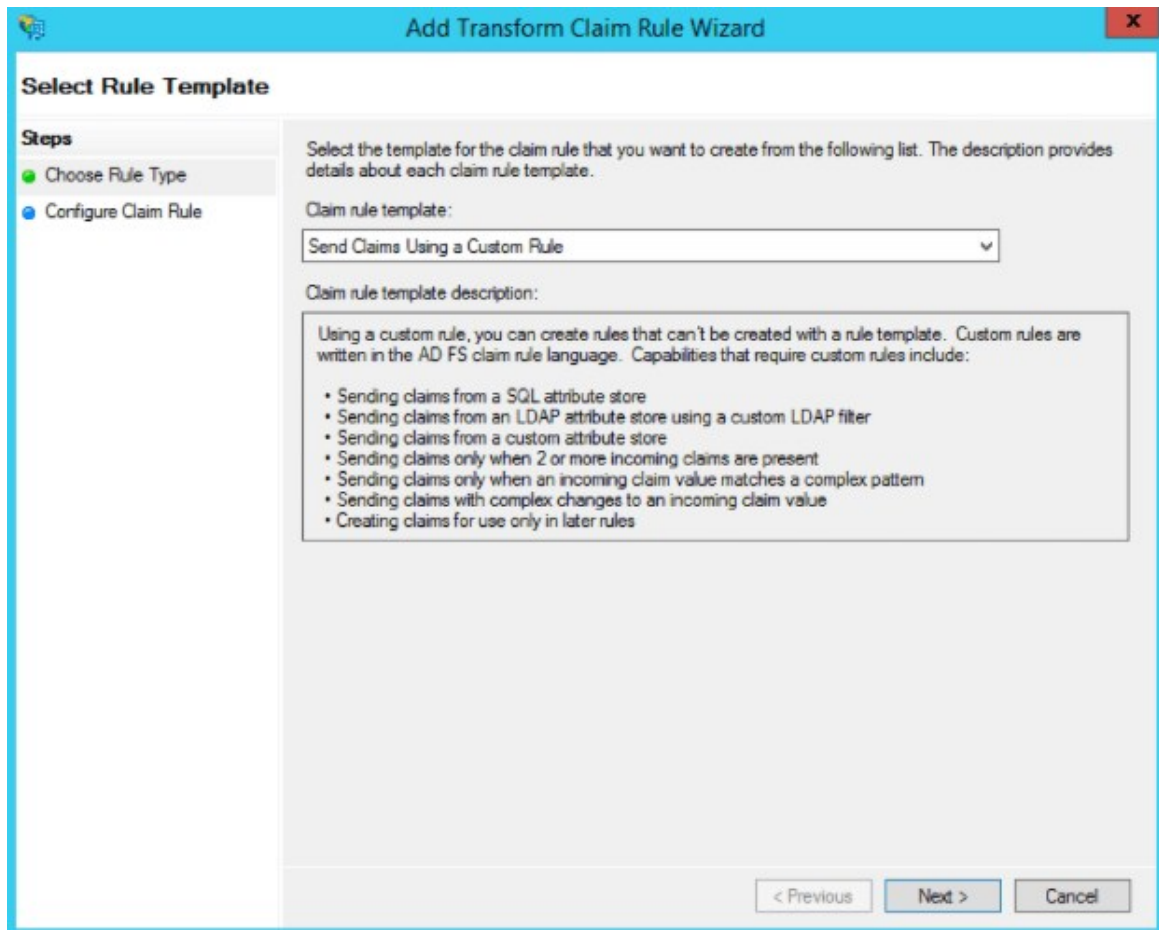
9. To save these configurations and use the default SAML assertion of <http://schemas.xmlsoap.org/claims/Group>, click **OK**. If you want to pass the claims assertion as a custom value “groups” in the SAML assertion, continue to the [Create Custom Value Groups](#) procedure below.

Create Custom Value “groups”

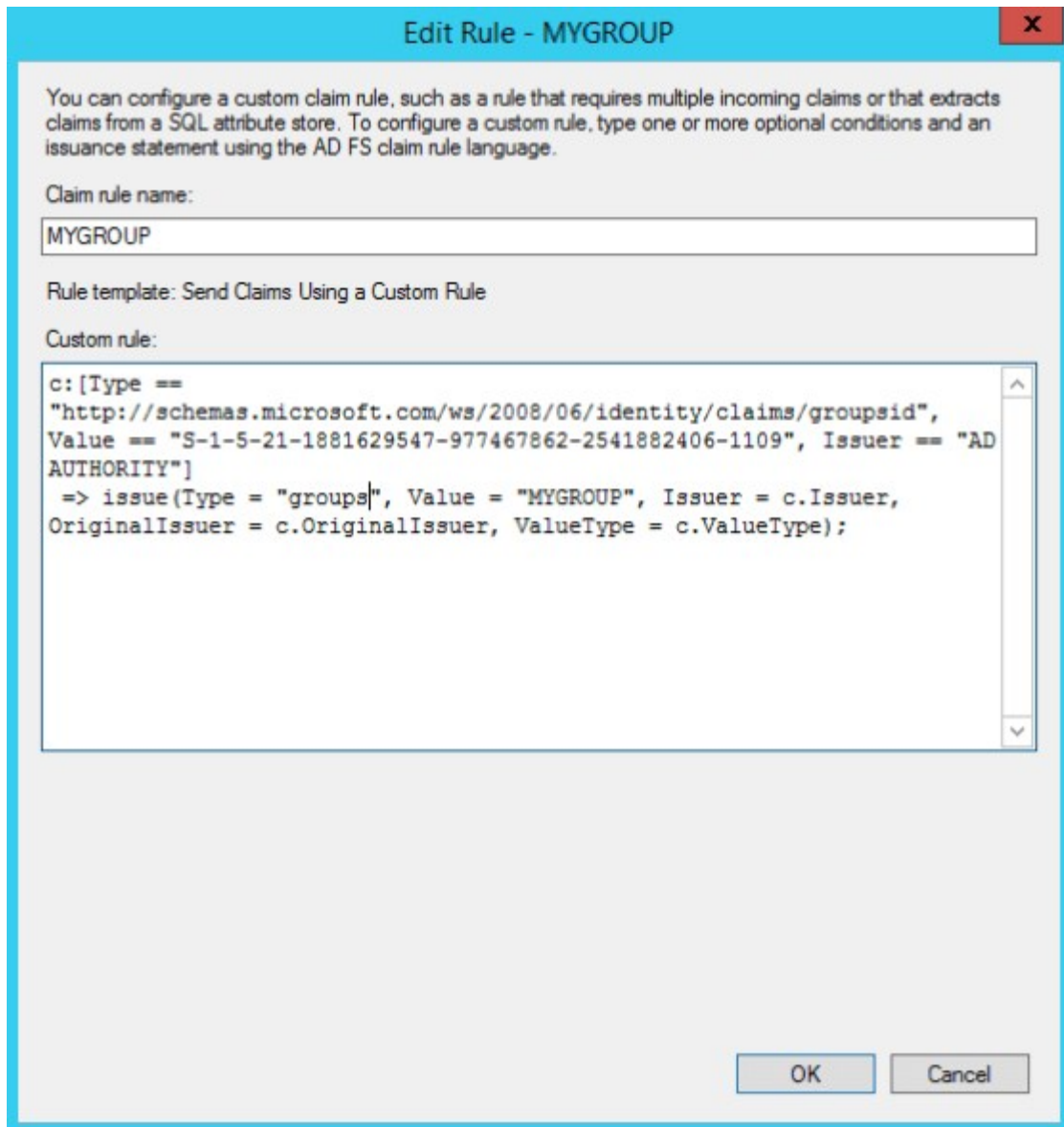
1. Select your newly created rule and click **Edit Rule**.



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** from the **Claim rule template** dropdown.
7. Click **Next**.



8. 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

9. 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: Pivotal Single Sign-On v1.11 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 Active Directory Federation Services (ADFS) external SAML identity provider to your Pivotal Single Sign-On service plan.

Overview

When you integrate ADFS with your deployment, you must configure an ADFS external SAML identity provider with specific settings. For information about how to add an external SAML identity provider in general, see [Add a SAML Provider](#).

To configure an ADFS external SAML identity provider, do the following:

1. [Download Identity Provider Metadata](#)
2. [Create a New SAML Identity Provider](#)
3. [Configure Your New Identity Provider](#)

Download Identity Provider Metadata

Download the metadata from your Active Directory Federation Services (ADFS) server at the following URL: <https://ADFS-HOSTNAME/federationmetadata/2007-06/federationmetadata.xml>

Create a New SAML Identity Provider

Follow the steps below to create a new identity provider:

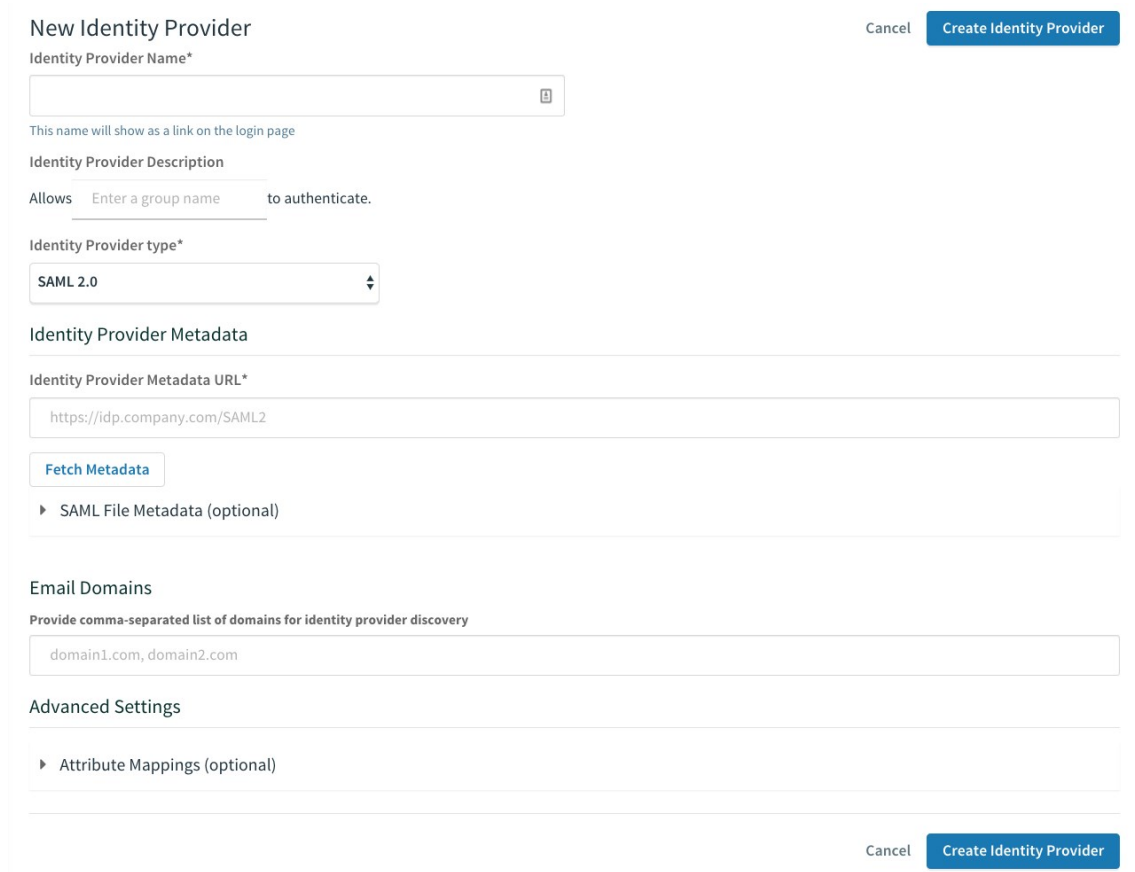
1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** from the dropdown.



Plans 1 **New Plan**

Name	Sign In Header
ADFS PCF SSO ▾	example
Edit Plan Manage Identity Providers	

- Click **New Identity Provider** to access configuration options.



New Identity Provider Cancel Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

Fetch Metadata

▸ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

Advanced Settings

▸ Attribute Mappings (optional)

Cancel Create Identity Provider

Configure Your New Identity Provider

Follow the steps below to configure a new identity provider:

- Enter an identity provider name below **Identity Provider Name**.
- (Optional) Enter a description under **Identity Provider Description**. This is displayed to space developers when they select an identity provider for their app.
- Select **SAML 2.0** from the dropdown under **Identity Provider Type**.
- Click **SAML File Metadata (optional)** and then click **Upload Identity Provider Metadata** to upload the XML metadata that you downloaded from your ADFS server. You do not need to enter a **Identity Provider Metadata URL**.



Note: Uploading the Identity Provider Metadata as an XML file makes you 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.

5. Enter the email domains you want to include as a comma-separated list under **Email Domains**.
6. Under **Advanced Settings**, click **Attribute Mappings (optional)**.
7. Configure **User Attributes** to determine how user attributes are propagated from the ADFS identity provider to Single Sign-On. For example, you might want to map ADFS' s SAML groups to Single Sign-On' s `external_groups`.
To map ADFS' s SAML groups to Single Sign-On' s `external_groups`, do the following:
 1. Under **User Schema Attribute**, select `external_groups`.
 2. If you followed the steps in [Create Custom Value Groups](#), type `groups` under **Attribute Name**. An attribute mapping with a customized SAML assertion value looks like this:

The screenshot shows the 'User Attributes' configuration interface. It has a title 'User Attributes' and a subtitle 'Map the incoming user attributes to known user schema.' Below this is a table with two columns: 'User Schema Attribute' and 'Attribute Name'. In the 'User Schema Attribute' column, there is a dropdown menu with 'external_groups' selected. In the 'Attribute Name' column, there is a text input field containing 'groups'. To the right of the input field is a plus sign icon.

If you did not follow the steps in [Create Custom Value Groups](#), type `http://schemas.xmlsoap.org/claims/group` under **Attribute Name**. An attribute mapping with a non-customized SAML assertion value looks like this:

The screenshot shows the 'User Attributes' configuration interface. It has a title 'User Attributes' and a subtitle 'Map the incoming user attributes to known user schema.' Below this is a table with two columns: 'User Schema Attribute' and 'Attribute Name'. In the 'User Schema Attribute' column, there is a dropdown menu with 'external_groups' selected. In the 'Attribute Name' column, there is a text input field containing 'http://schemas.xmlsoap.org/claims/Group'. To the right of the input field is a plus sign icon.

8. Click **Create Identity Provider**.
9. From the identity provider list, click on the name of the ADFS identity provider and then click **Group Whitelist**.
10. Follow the instructions in [Configure Group Allowlist for an External Identity Provider](#) to enter the ADFS SAML group names to be propagated in the ID tokens generated by Single Sign-On. These SAML groups are now included in the roles claim of the user' s ID token.
11. From the identity provider list, click on the name of the ADFS identity provider and then click **Resource Permissions**.
12. Follow the instructions in [Create or Edit Resource Permissions Mapping](#) to map the user' s ADFS group memberships to their access token' s scopes. The resource permissions that the SAML groups were mapped to are now included in the scopes claim of the user' s access token.

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

Testing



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and Active Directory Federation Services (AD FS). An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

Test Your Service Provider Connection

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

Overview


Settings

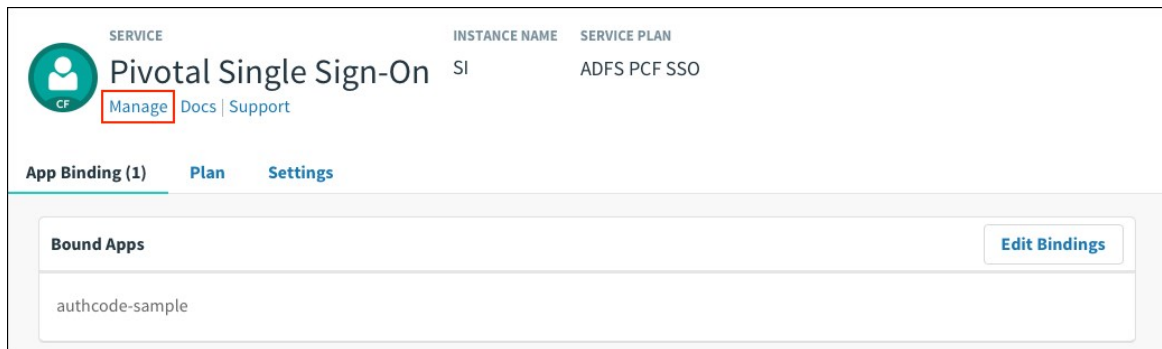
Apps

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

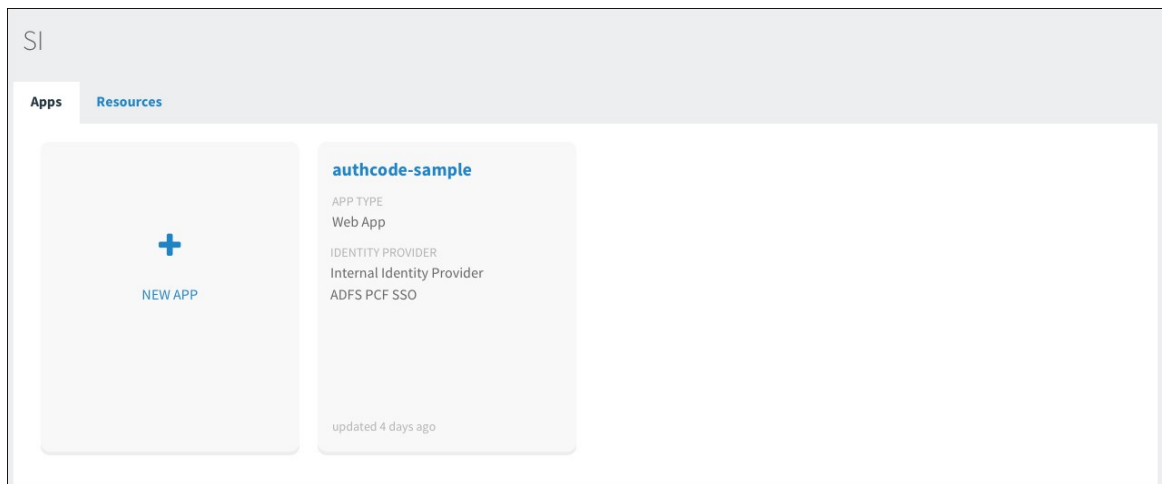
Services

Add Service

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



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



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 app 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**.

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

This is the Access Token that was used:

```
{
  "jti" : "bbe64fd09cbf4ed4a4fdf17c3ea8af04",
  "sub" : "5651953a-c864-4073-86b3-4fa97b45a0d1",
  "scope" : [ "todo_read", "openid", "todo_write" ]
}
```



```

    "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?

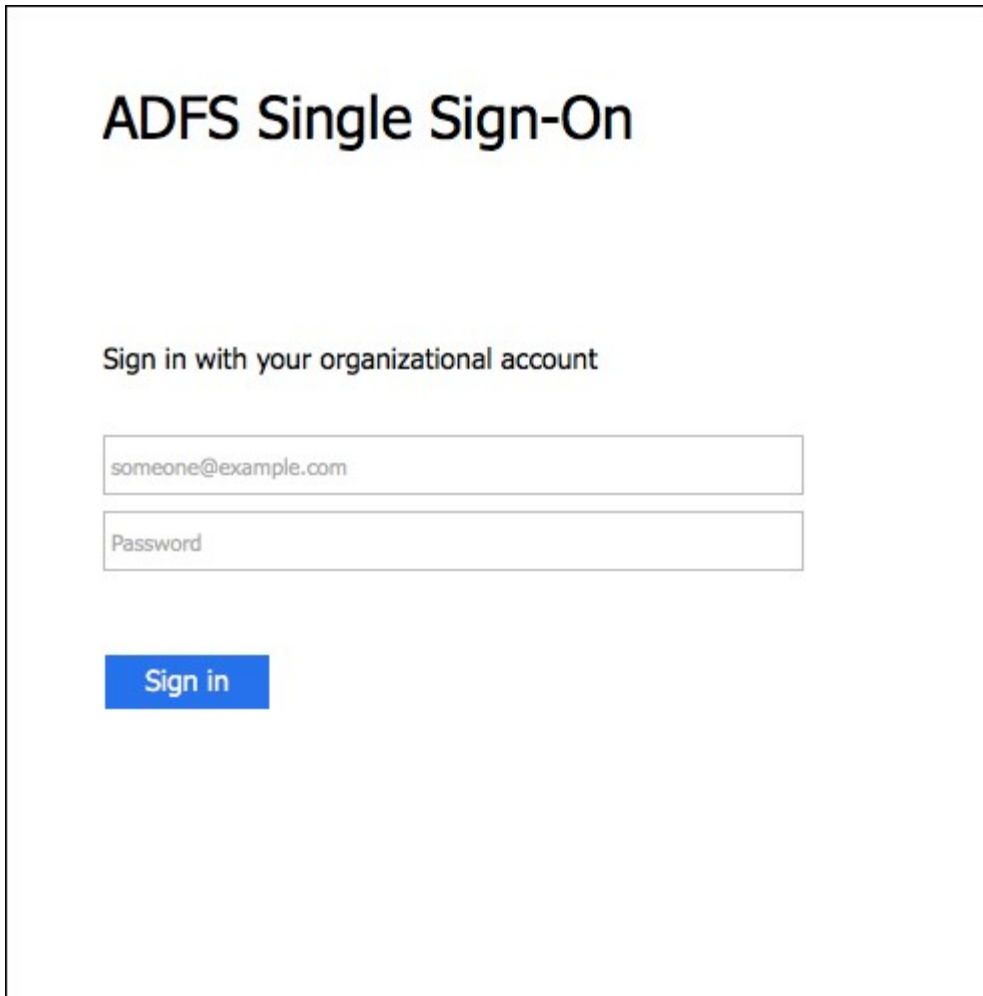
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



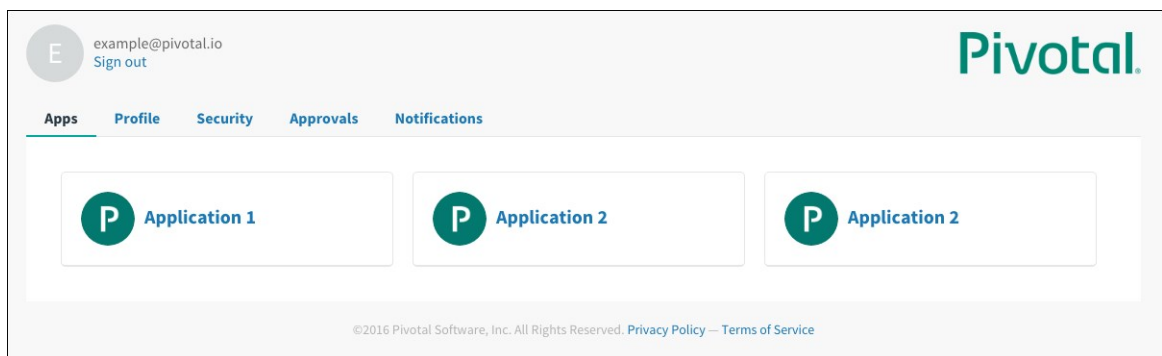
Note: Single Sign-On does not support identity provider-initiated flow into app, but it does redirect the user to the User Account and Authentication (UAA) page to select apps assigned to the user.

1. Sign in to AD FS.



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

2. Navigate to your app and click it.
3. You are redirected to the page that lists apps 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 app. 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

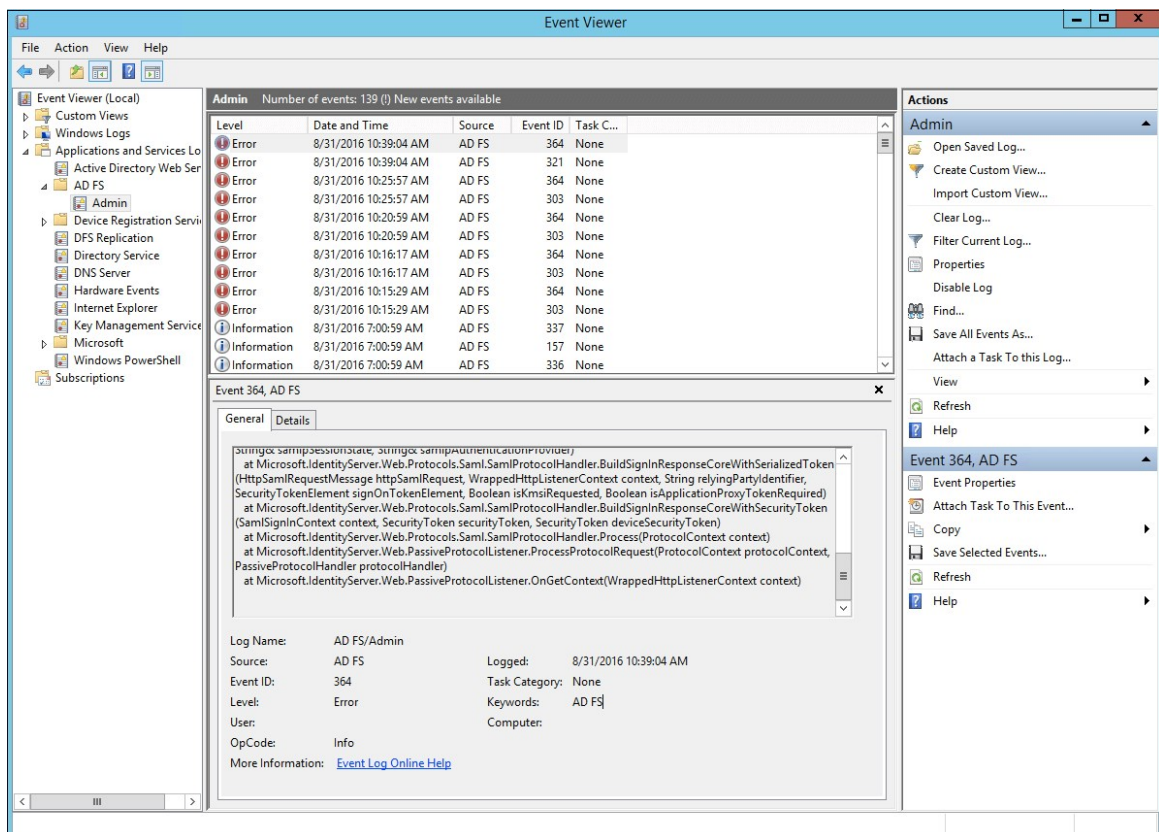


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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.

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 Overview



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On.

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 Platform, you must have the following:

- An Azure AD 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 Single Sign-On

Complete both steps below to integrate your deployment with Azure AD and Single Sign-On.

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: Pivotal Single Sign-On v1.11 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.

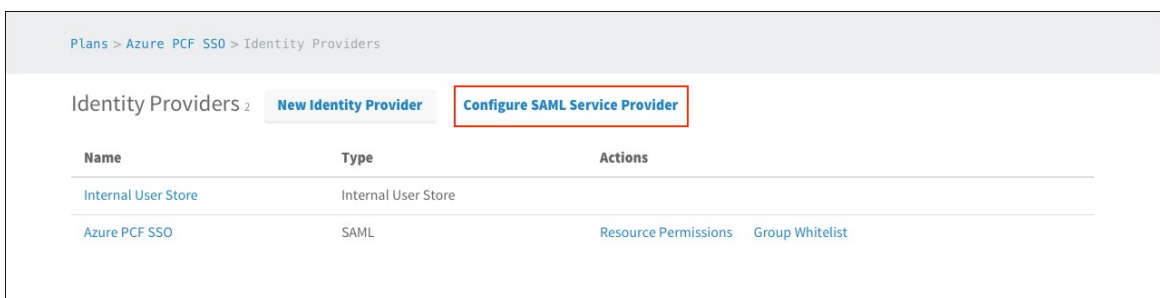
This topic describes how to set up Azure Active Directory (AD) as your identity provider by configuring SAML integration in both Pivotal Single Sign-On and Azure AD.

Step 1: Set up SAML in Single Sign-On

1. Log in to the SSO Operator Dashboard at <https://p-identity.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

[Download Metadata](#)☒ Perform signed authentication requests☐ Require signed assertions

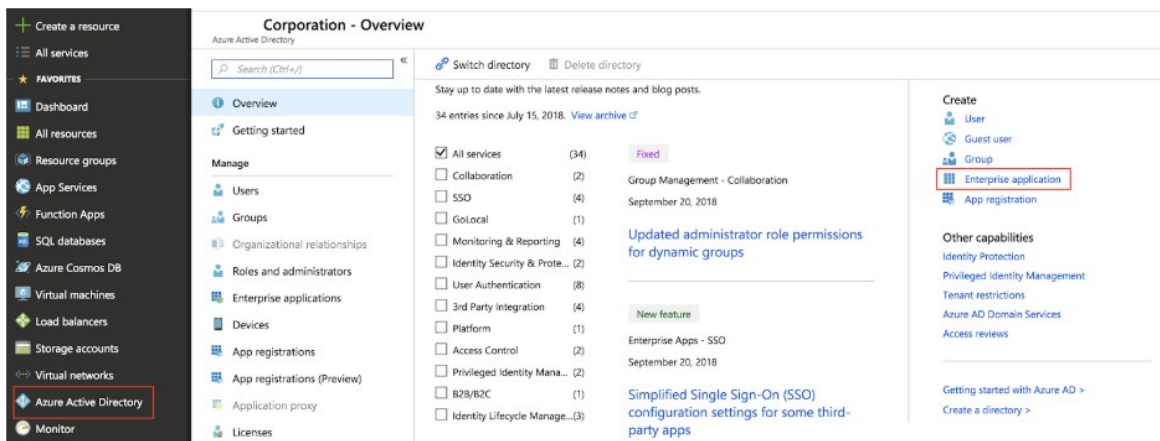
Cancel

Save

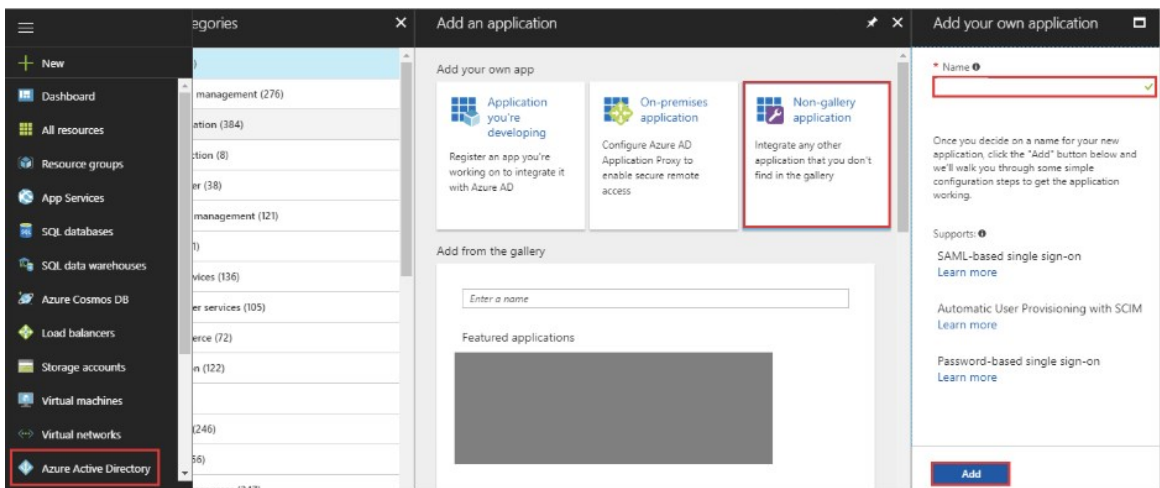
- (Optional) Select **Require signed assertions** to validate the origin of signed responses.
- Click **Download Metadata** to download the service provider metadata.
- Click **Save**.

Step 2: Set up SAML in Azure AD

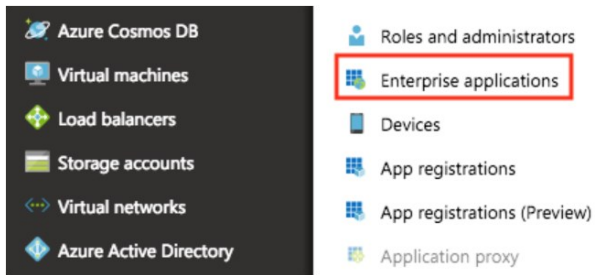
- Log in to Azure AD as a Global Administrator at <https://portal.azure.com/>.
- Navigate to **Azure Active Directory** tab > **Enterprise application**.



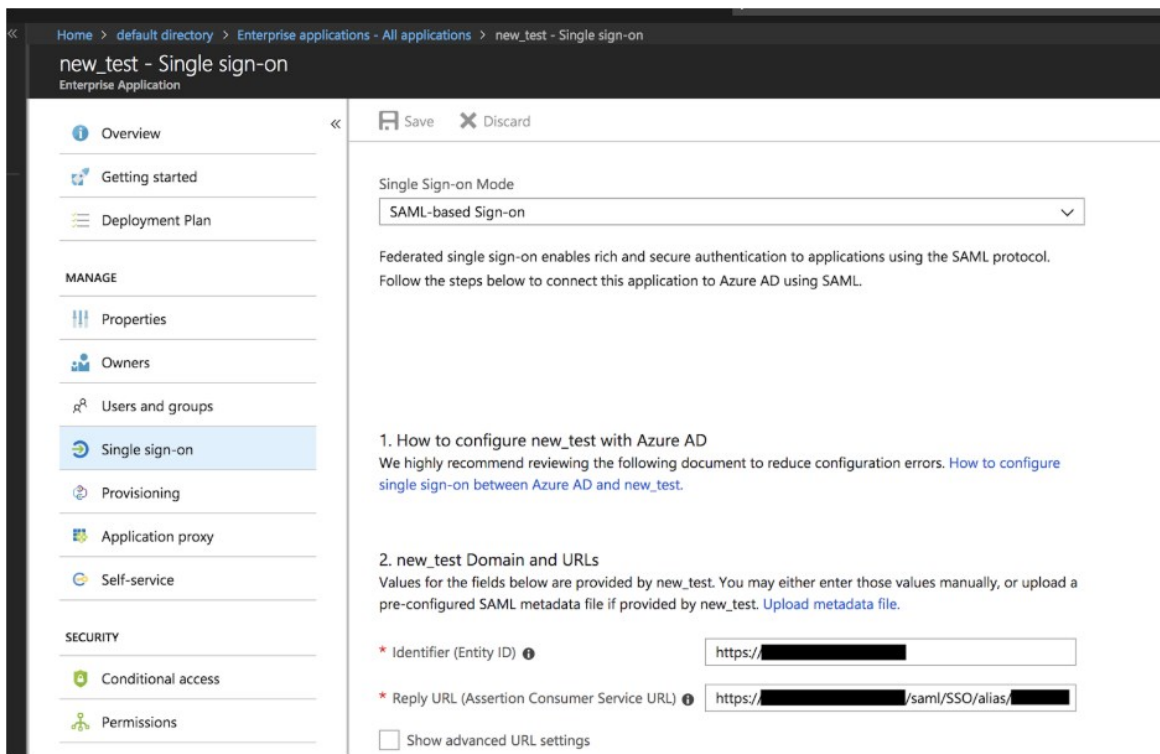
- Select **Non-gallery application**. Provide a name and click **Add**.



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



5. Click your app 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 Single Sign-On](#).



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 

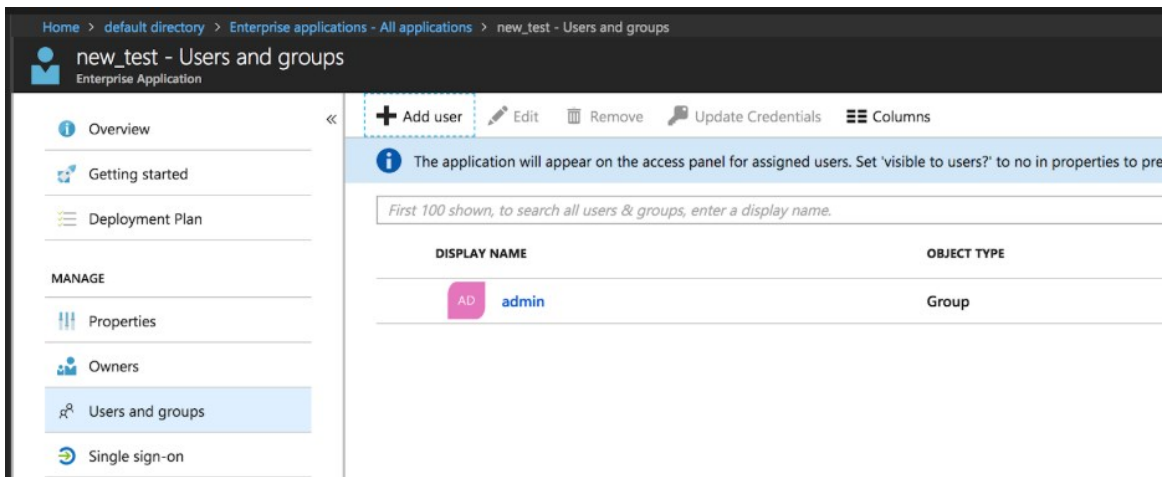
STATUS	EXPIRATION	THUMBPRINT	DOWNLOAD
Active	8/16/2021	1534B2F28AA4563EBD7AA9A15AC23767FD32941D	Certificate (Base64) Certificate (Raw) Metadata XML

[Create new certificate](#)

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

* Notification Email 

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



Home > default directory > Enterprise applications - All applications > new_test - Users and groups

new_test - Users and groups
Enterprise Application

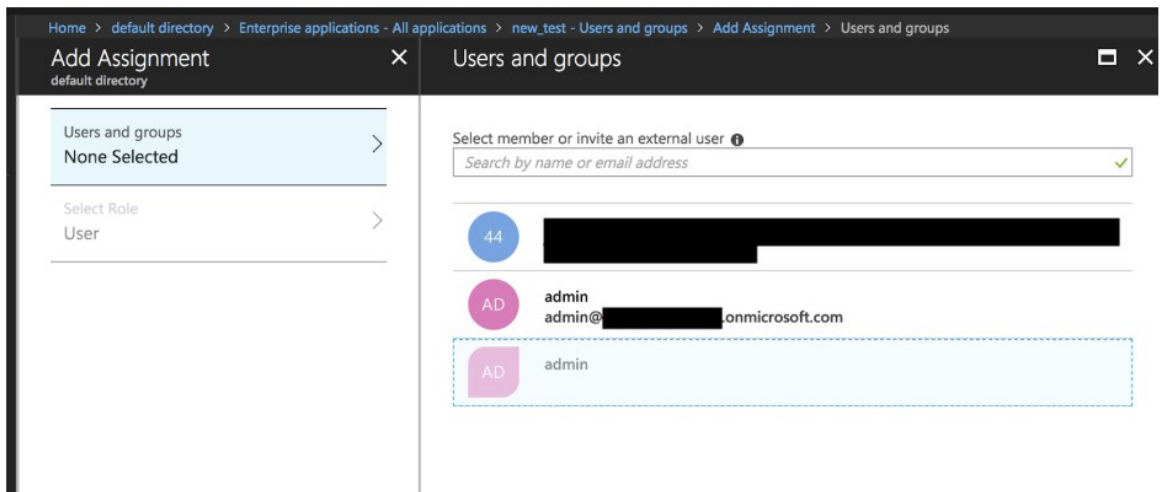
Overview | **+ Add user** | Edit | Remove | Update Credentials | Columns

The application will appear on the access panel for assigned users. Set 'visible to users?' to no in properties to pre

First 100 shown, to search all users & groups, enter a display name.

DISPLAY NAME	OBJECT TYPE
AD admin	Group

- Select users or group names from the dropdown. For example, you can add a group that includes all users that should be able to log in to the Single Sign- On plan.





Home > default directory > Enterprise applications - All applications > new_test - Users and groups > Add Assignment > Users and groups

Add Assignment
default directory

Users and groups
None Selected

Select Role
User

Select member or invite an external user 

Search by name or email address 

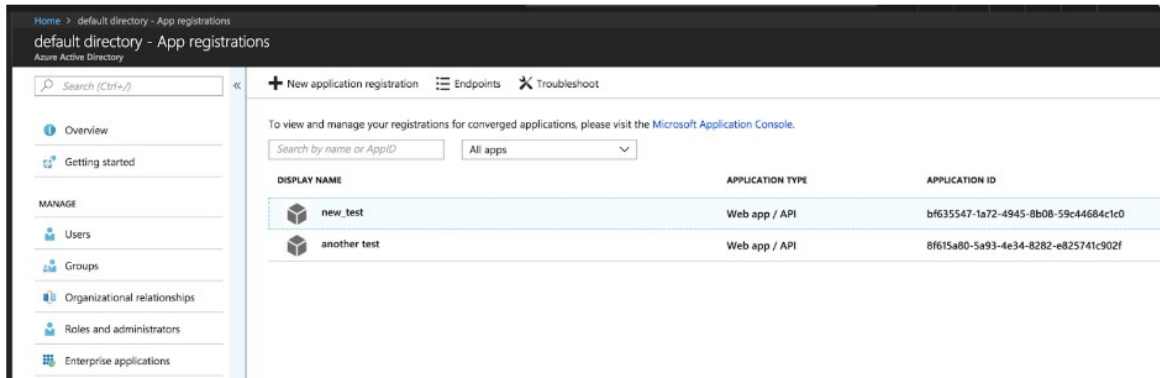
44 [redacted]

AD admin
admin@[redacted]onmicrosoft.com

AD admin

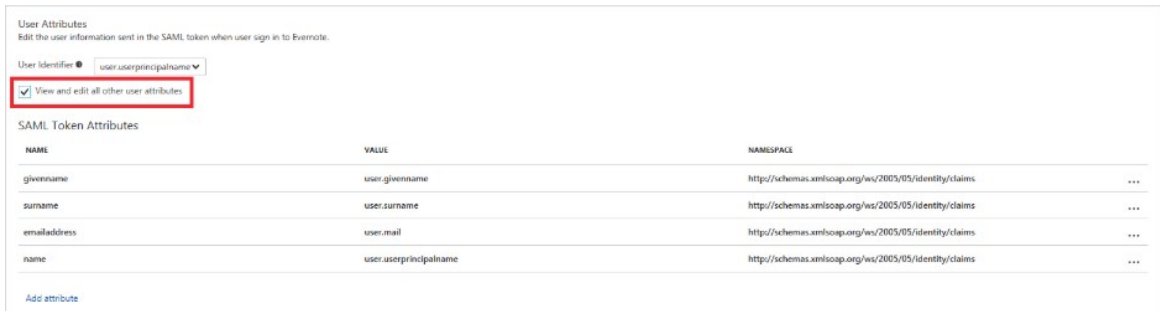
Step 3: Set up Claims Mapping

1. Navigate to **Azure Active Directory > App registrations**. Click your app.



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 the [Microsoft documentation](#).



[View a larger version of this image.](#)

3. To pass group membership claims to the app:
 1. Click **Manifest**.
 2. Locate `groupMembershipClaims` and set the value to one of the following:
 - `SecurityGroup`. Groups claim contains identifiers of all security groups of which the user is a member.
 - `All`. Groups claim contains the identifiers of all security groups and distribution lists of which the user is a member.
 3. Save the change.

For more information, see the [Microsoft documentation](#).

Edit manifest

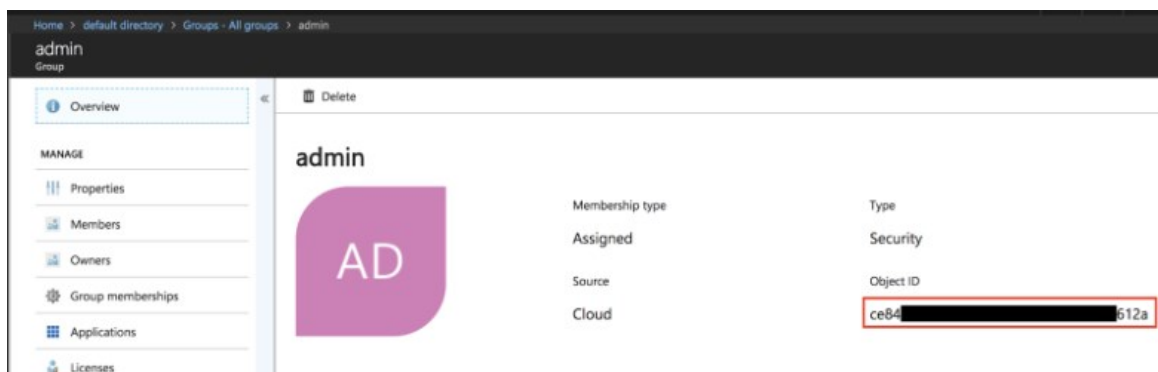
Save Discard Edit Upload Download

```

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 "passwordClaims": null

```

4. Navigate to **Azure Active Directory > Groups**.
5. For each group that is used by the Single Sign-On plan, record the **Object ID**. Azure AD passes the Object ID of these groups to the Single Sign-On 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: Pivotal Single Sign-On v1.11 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 service plan.

Step 1: Set up SAML

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



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

New Identity Provider

Cancel **Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

<https://idp.company.com/SAML2>

Fetch Metadata

▶ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

Advanced 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 passes the Object ID of the groups recorded in step 5 of [Step 3: Set up Claims Mapping](#) to the Single Sign- On 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 Operator Dashboard at <https://p-identity.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 Single Sign-On by following these steps:
 1. Log in at <https://p-identity.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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and Azure Active Directory (AD). An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

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 app 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.SYSTEM-DOMAIN> and navigate to the org and space where your application is located.
2. Under **Services**, locate the service instance of the Single Sign-On plan bound to your app. Click on the service instance 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 Azure PCF SSO

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

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

Bound Apps [Edit Bindings](#)

authcode-sample

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

SI

Apps **Resources**

authcode-sample

APP TYPE
Web App

IDENTITY PROVIDER
Internal Identity Provider
Azure PCF SSO

updated 4 days ago

[NEW APP](#)

- 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 app 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.

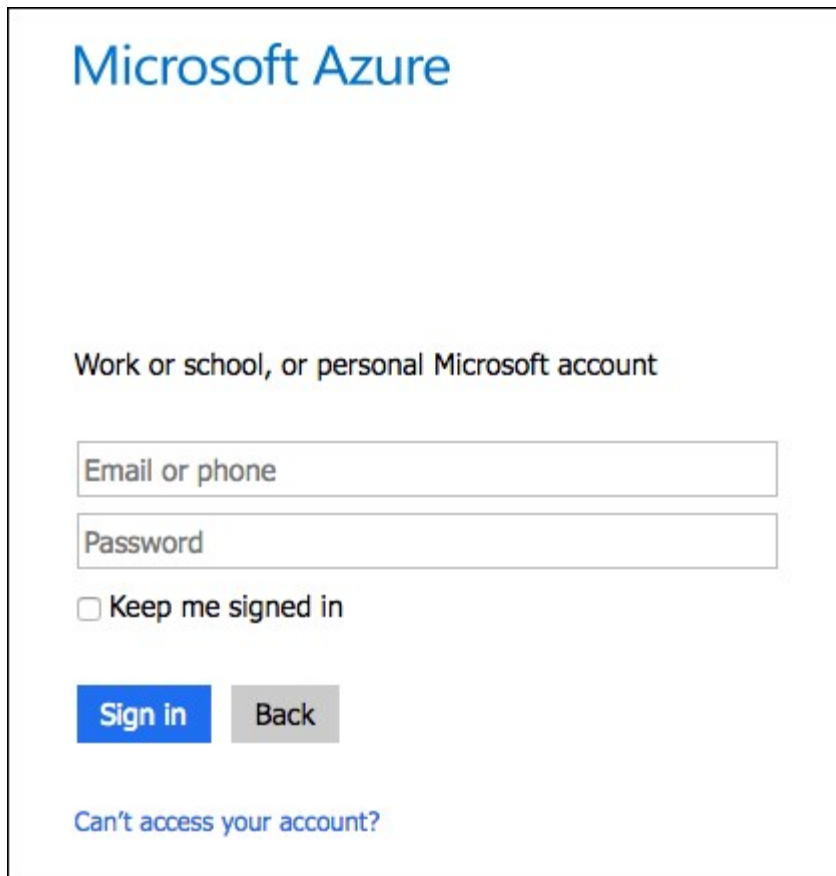
← → ↺ https://authcode-sample

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

8. The app 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?

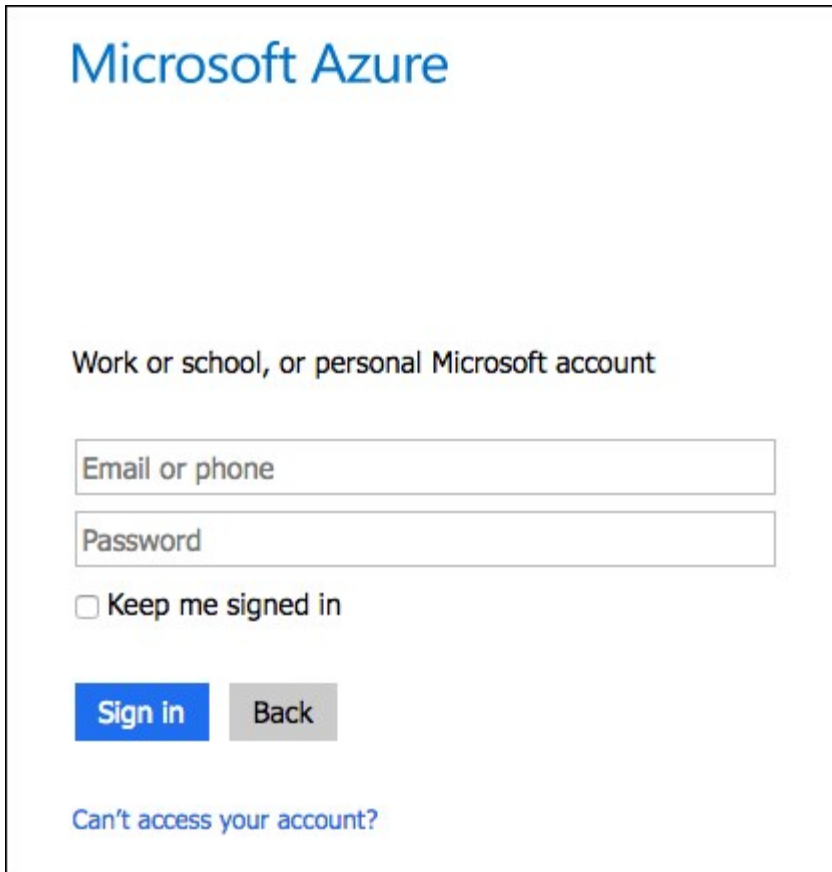
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



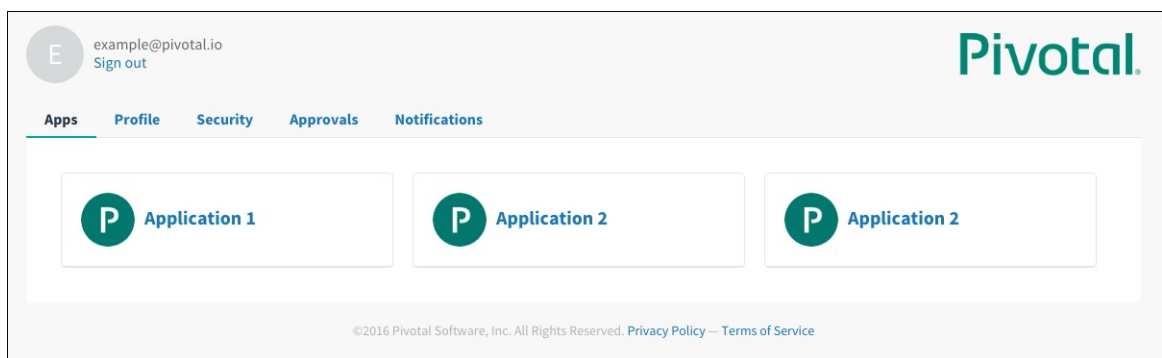
Note: Single Sign-On does not support identity provider-initiated flow into applications, but it does redirect the user to the User Account and Authentication (UAA) page to select apps assigned to the user.

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?".

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



Test Your Single Sign-Off

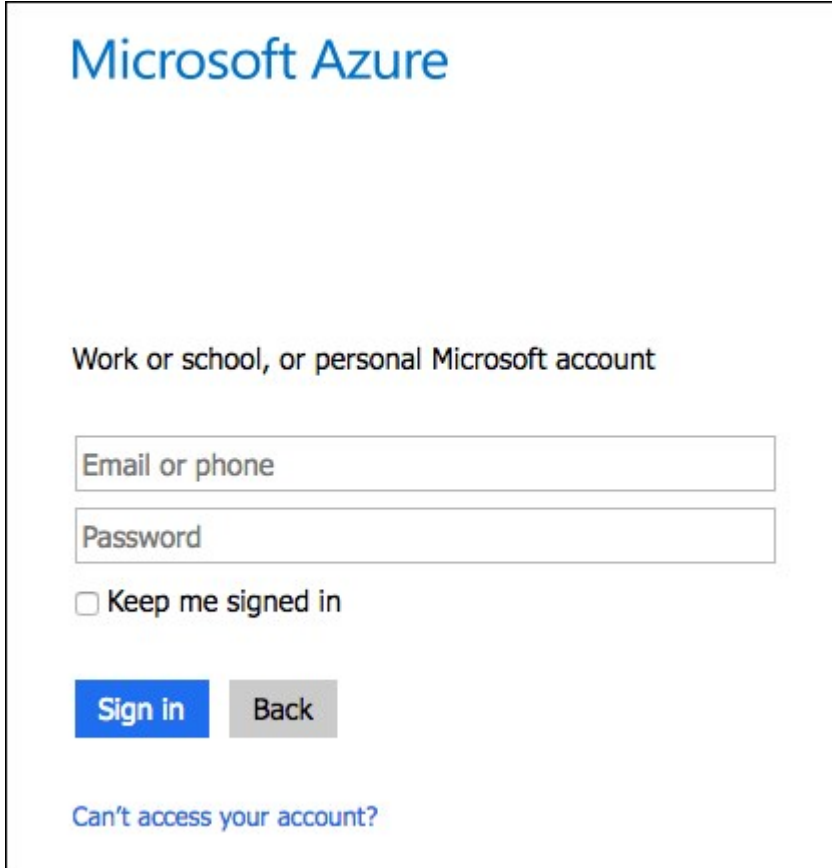
Test single sign-off to ensure that when users log out of the application, they are logged out of Azure AD as well.

1. Sign into the sample app. 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.



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?](#)

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

Troubleshooting



Warning: Pivotal Single Sign-On v1.11 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.

Failed Login

Symptom:

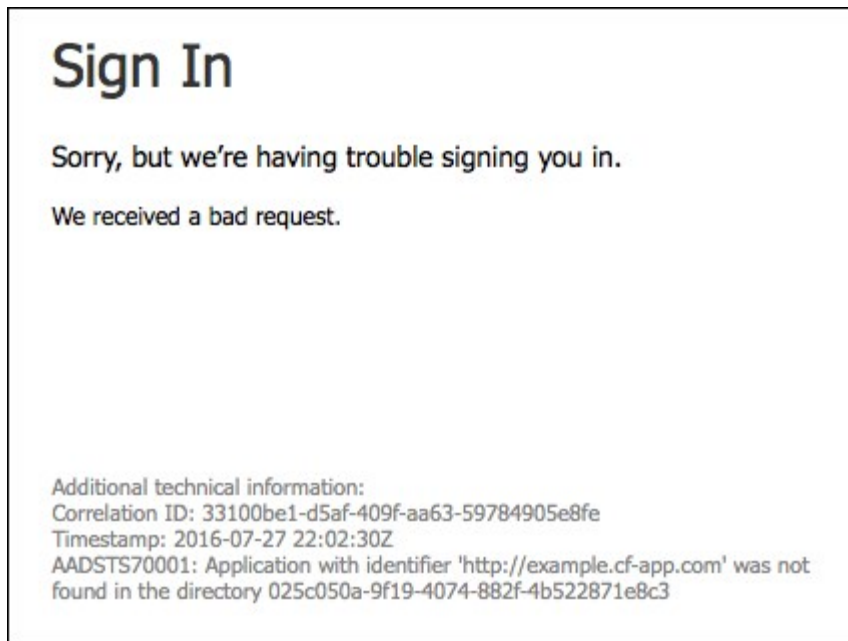
You cannot log in to your Single Sign-On plan.

Solutions:

- Pivotal recommends using a different browser or deleting your browser cache and history before you log in to your Single Sign-On plan. Your Single Sign-On 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 Single Sign-On 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:

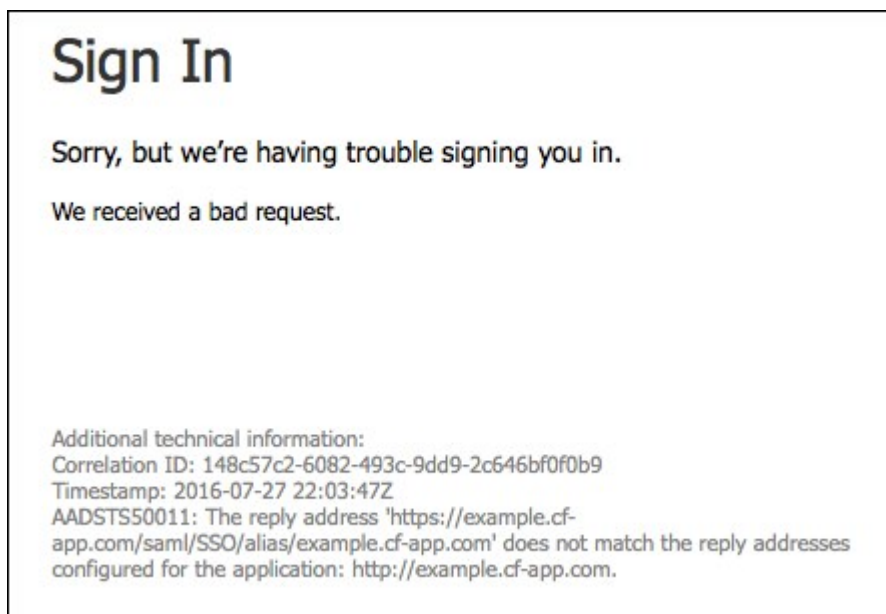


Explanations:

- The App ID URI is misconfigured on Azure AD.

Reply URL Does Not Match

Symptom:



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 Overview



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On.

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 using OIDC, you must have the following:

- 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 Single Sign-On

To integrate your deployment with Azure AD and Single Sign-On, follow the steps in [Configure Azure AD as an OIDC Identity 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On plan, by configuring OpenID Connect (OIDC) in both Single Sign-On and Azure AD.

Overview

Azure Active Directory (Azure AD) is Microsoft's multi-tenant cloud-based directory and identity management service. It is one of several identity providers you can use in a Single Sign-On service plan.

To set up the integration, follow the procedures below:

1. [Set up a Relying Party in Azure AD](#)
2. [Set up the OIDC Identity Provider in Single Sign-On](#)

Prerequisites

Before you can set up a relying party in Azure AD, you must meet the prerequisites listed in [Azure Active Directory OIDC Integration Guide Overview](#).

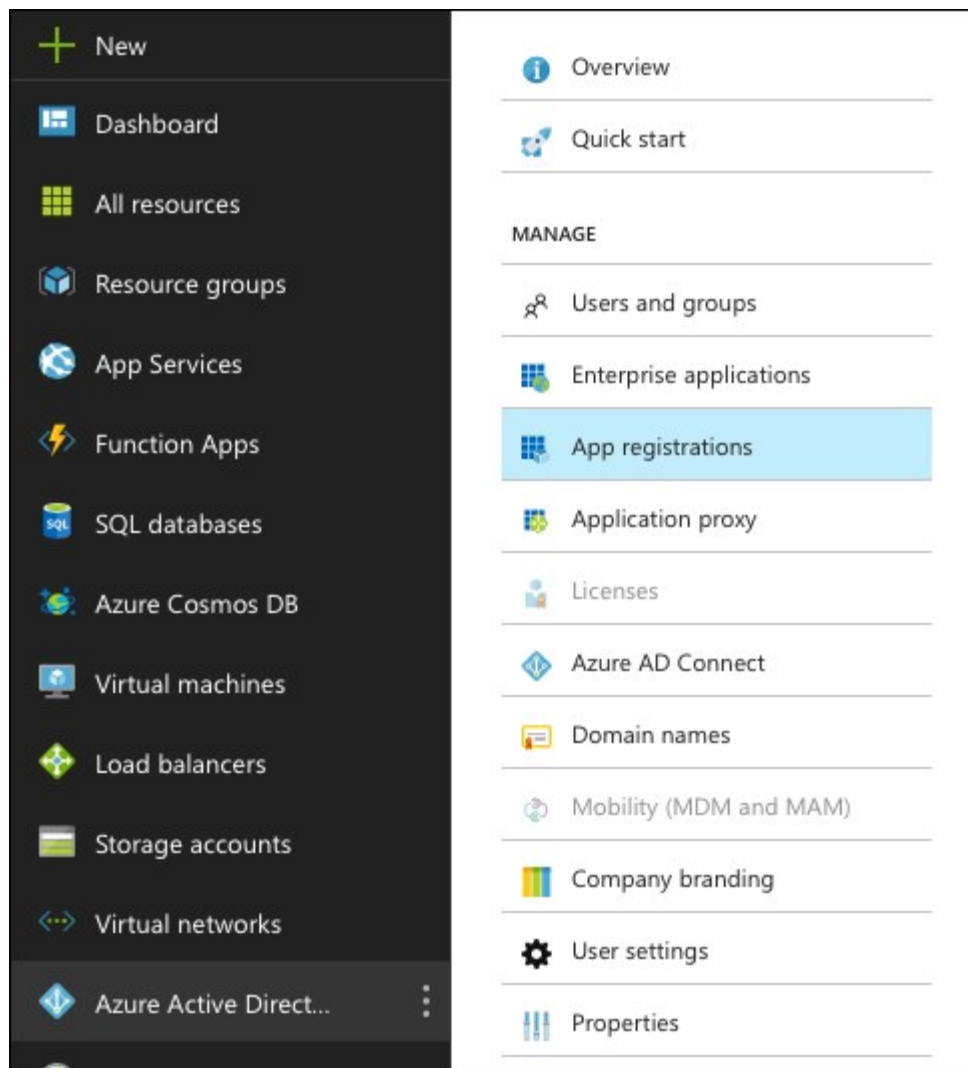
Set Up a Relying Party in Azure AD

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

Register a New App

To register a new app:

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.

+ New application registration
 ☰ Endpoints
✕ Troubleshoot

To view and manage your registrations for converged applications, please visit the [Microsoft Application Console](#).

All apps
 ▼

3. Enter a name of your choice in the **Name** field.
4. Select **Web App/API** from the **Application type** dropdown.
5. Enter the sign-on URL in the **Sign-on URL** field. You can use the URL for the login portal, if you want. This URL has the following pattern:

`https://AUTH-DOMAIN.login.SYSTEM-DOMAIN`

For example:

The 'Create' dialog box contains the following fields:

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

Generate a Relying Party OAuth Client Secret

To create a client secret:

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

The screenshot shows the 'New application registration' page with a table of registered applications:

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

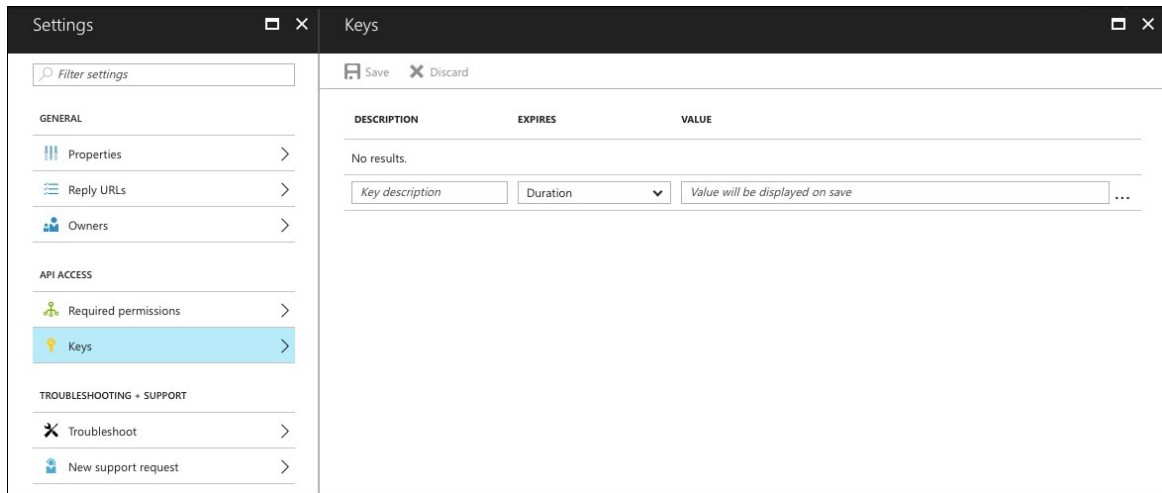
2. Record the **Application ID** displayed on the screen. This is the **Relying Party OAuth Client ID**.

The 'Example OIDC Client' settings page displays the following information:

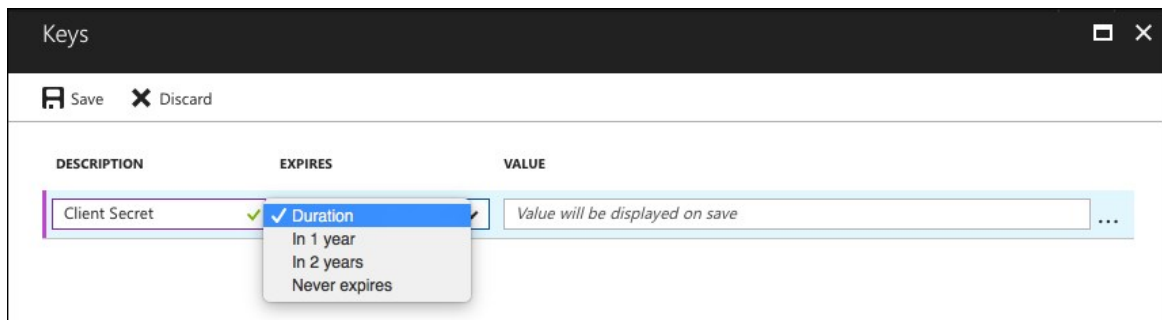
- Display name:** Example OIDC Client
- Application type:** Web app / API
- Home page:** https://example.login.mydomain.org
- Application ID:** [Redacted]
- Object ID:** [Redacted]
- Managed application in local directory:** Example OIDC Client

Buttons at the top include Settings, Manifest, and Delete. An 'All settings' button is at the bottom right.

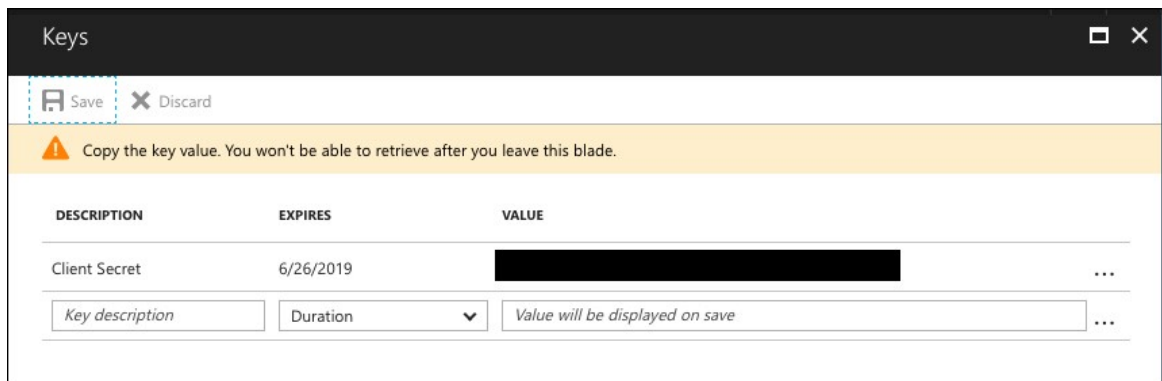
3. Open the **Keys** tab.



4. Enter a name for the key in the **DESCRIPTION** field.
5. Select a duration appropriate for your security requirements in the **EXPIRES** field.



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



Configure Reply and Endpoint URLs

To create reply and endpoint URLs:

1. Under **Reply URLs**, configure and save the URL using the following pattern:

```
https://AUTH-DOMAIN.login.SYSTEM-DOMAIN/login/callback/ORIGIN-KEY
```

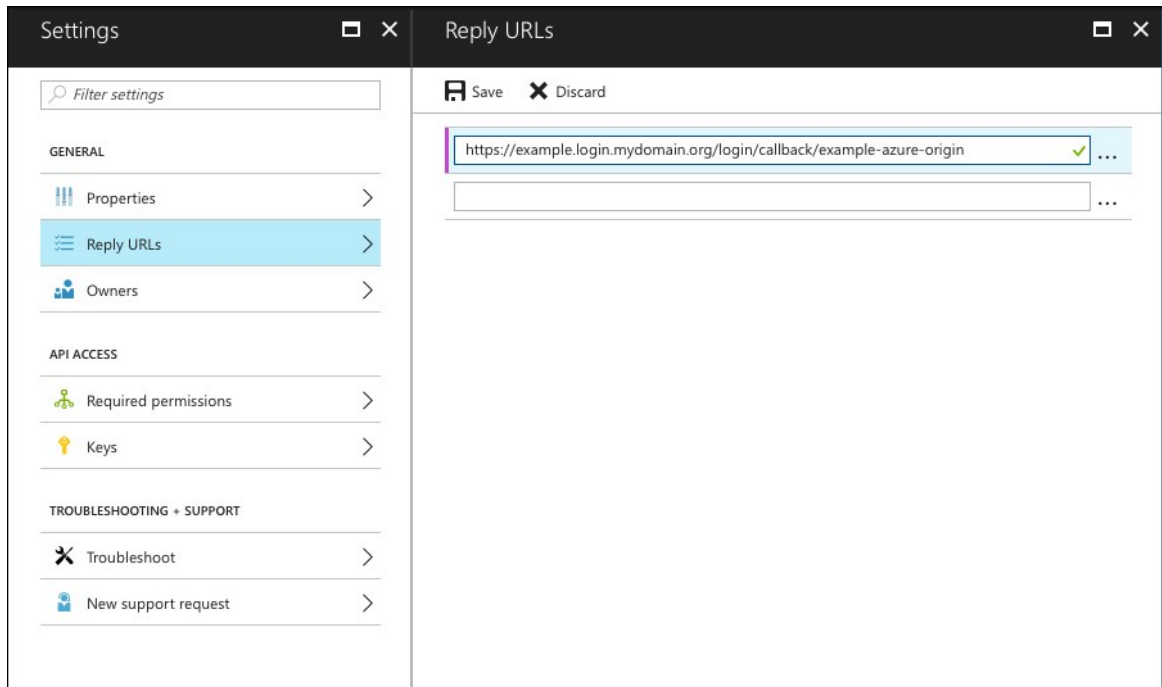
Where:

- ♦ **AUTH-DOMAIN** is the **Auth Domain** you entered in [Create or Edit Service Plans](#).

- ✦ **ORIGIN-KEY** is based on the **Identity Provider Name** you set in the SSO Operator Dashboard in **Set Up OIDC Identity Provider in SSO** as shown below. Do not use spaces or uppercase letters in this value.



Warning: The origin key does not change after it is assigned, even if the **Identity Provider Name** is modified.



2. 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 a management console. The left window, titled 'Settings', has a sidebar with a search bar and several categories: 'GENERAL' (containing 'Properties', 'Reply URLs', and 'Owners'), 'API ACCESS' (containing 'Required permissions' and 'Keys'), and 'TROUBLESHOOTING + SUPPORT' (containing 'Troubleshoot' and 'New support request'). The 'Properties' option is selected and highlighted in blue. The right window, titled 'Properties', has a 'Save' button and a 'Discard' button. It contains the following fields: 'Name' (Example OIDC Client), 'Object ID' (redacted), 'Application ID' (redacted), 'App ID URI' (https://[redacted].onmicrosoft.com/[redacted], with the domain part highlighted by a red box), 'Logo' (a blue square with 'EO'), 'Upload new logo' (a file selection button), 'Home page URL' (https://example.login.mydomain.org), 'Logout URL' (empty), 'Application type' (Web app / API), and 'Multi-tenanted' (Yes/No buttons, with 'No' selected).

- 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 Single Sign-On.

```
{
  "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"
  ],
  "micro soft 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"
  ],
  "micro soft 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"
}
```

Set Up the OIDC Identity Provider in Single Sign- On

Follow the steps below to set up an OIDC provider for Single Sign- On:

1. Follow steps 1 – 6 in [Add an OIDC Provider](#).
2. Clear the **Enable Discovery** checkbox and enter the following information from the **OpenID Connect metadata endpoint** you constructed in the final step of [the previous section](#).

For...	Do the following...
Authorization Endpoint URL	Enter the <code>authorization_endpoint</code> value from the metadata endpoint.
Token Endpoint URL	Enter the <code>token_endpoint</code> value from the metadata endpoint.
Token Key	Enter the <code>jwks_uri</code> value from the metadata endpoint.
Issuer	Enter the <code>issuer</code> value from the metadata endpoint.
User Info Endpoint URL	Enter the <code>userinfo_endpoint</code> value from the metadata endpoint.
Response Type	Select <code>code</code> from the dropdown.

Relying Party OAuth Client ID	Enter the Application ID you recorded in step 5 of Configuring Azure Active Directory as an OIDC Identity Provider .
Relying Party OAuth Client Secret	Enter the Client Secret you recorded in step 8 of Configuring Azure Active Directory as an OIDC Identity Provider .

3. Select `openid` as a scope. You can select additional scopes.
4. Under **Advanced Settings > Attribute Mappings (optional) > User Attributes**, select `user_name` as the **User Schema Attribute** and enter `unique_name` as the **Attribute Name**. This enables Single Sign-On to identify the authenticated user.
5. (Optional) Configure additional attribute mappings.
6. Click **Create Identity Provider** to save your settings.
7. (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



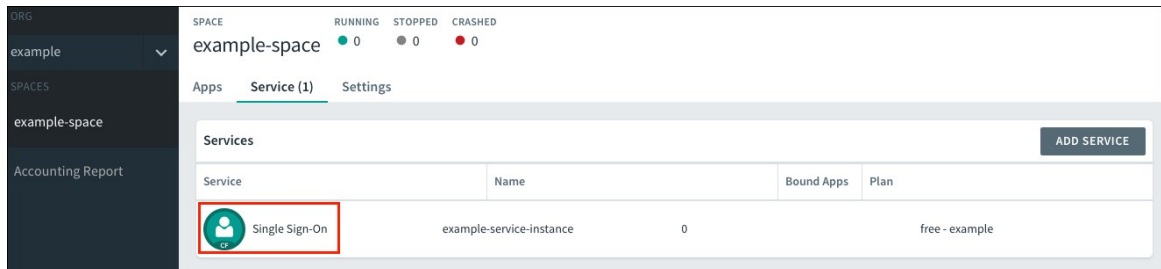
Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how a Pivotal Platform administrator can test the OpenID Connect (OIDC) connection between Pivotal Single Sign-On and Azure Active Directory (Azure AD).

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

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

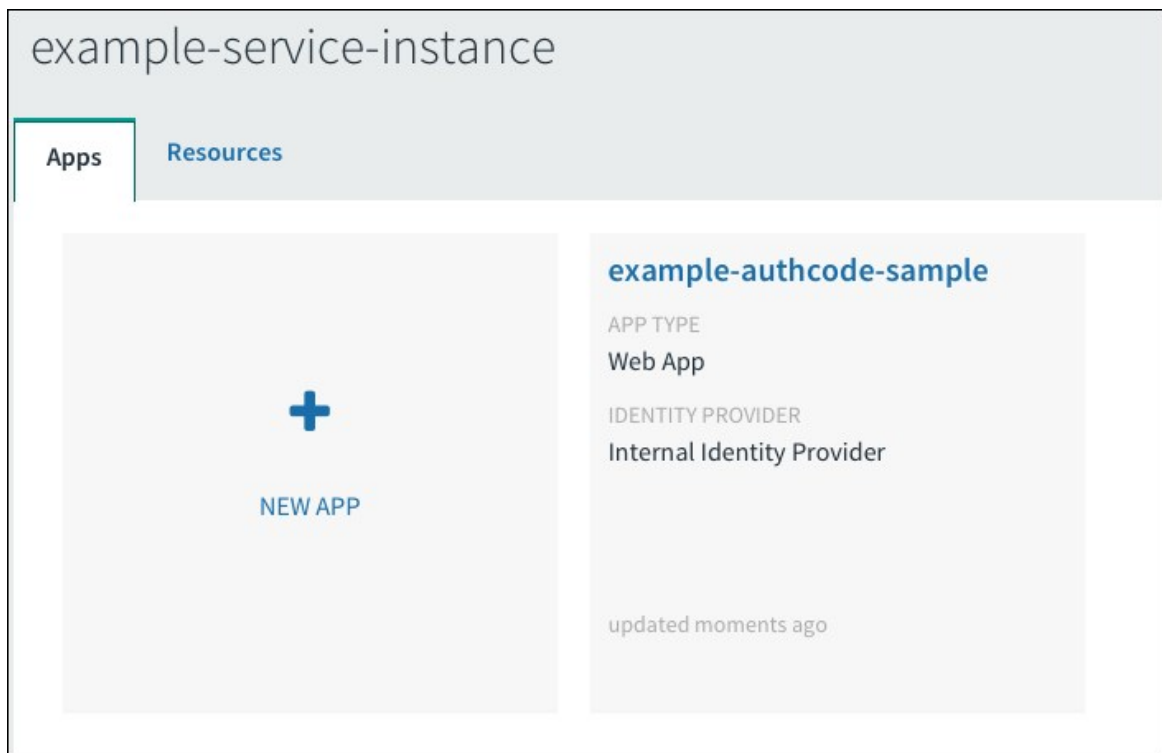
1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN> and navigate to the org and space where your app is located.
2. Under **Services**, locate the service instance of the Single Sign-On plan bound to your app.



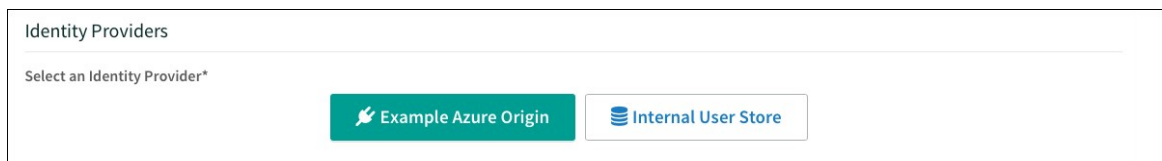
3. Select the service instance and click **Manage**.



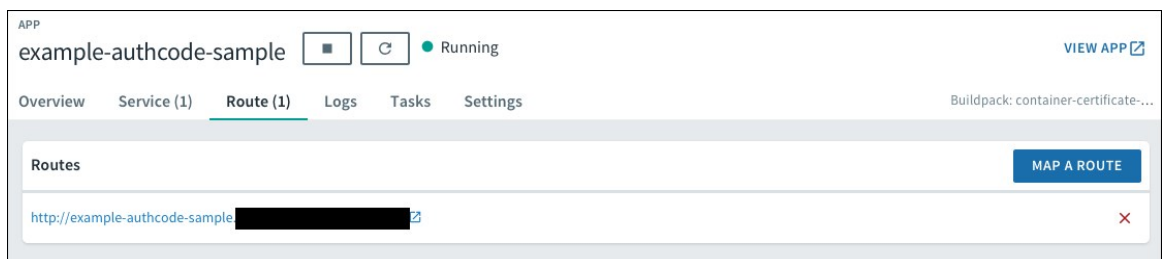
4. Under the **Apps** tab, select your app.



- 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 Single Sign- On 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: Pivotal Single Sign-On v1.11 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.

Bad Request

Symptom:



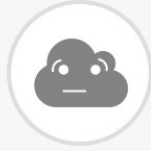
There was an error when authenticating against the external identity provider: 400 Bad Request

Explanations:

- This is a generic error. Review UAA logs for detailed information.
- This error can occur when the app 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 `code` is used. Ensure you configure the response type to use `code`.

Cannot determine username from credentials supplied

Symptom:



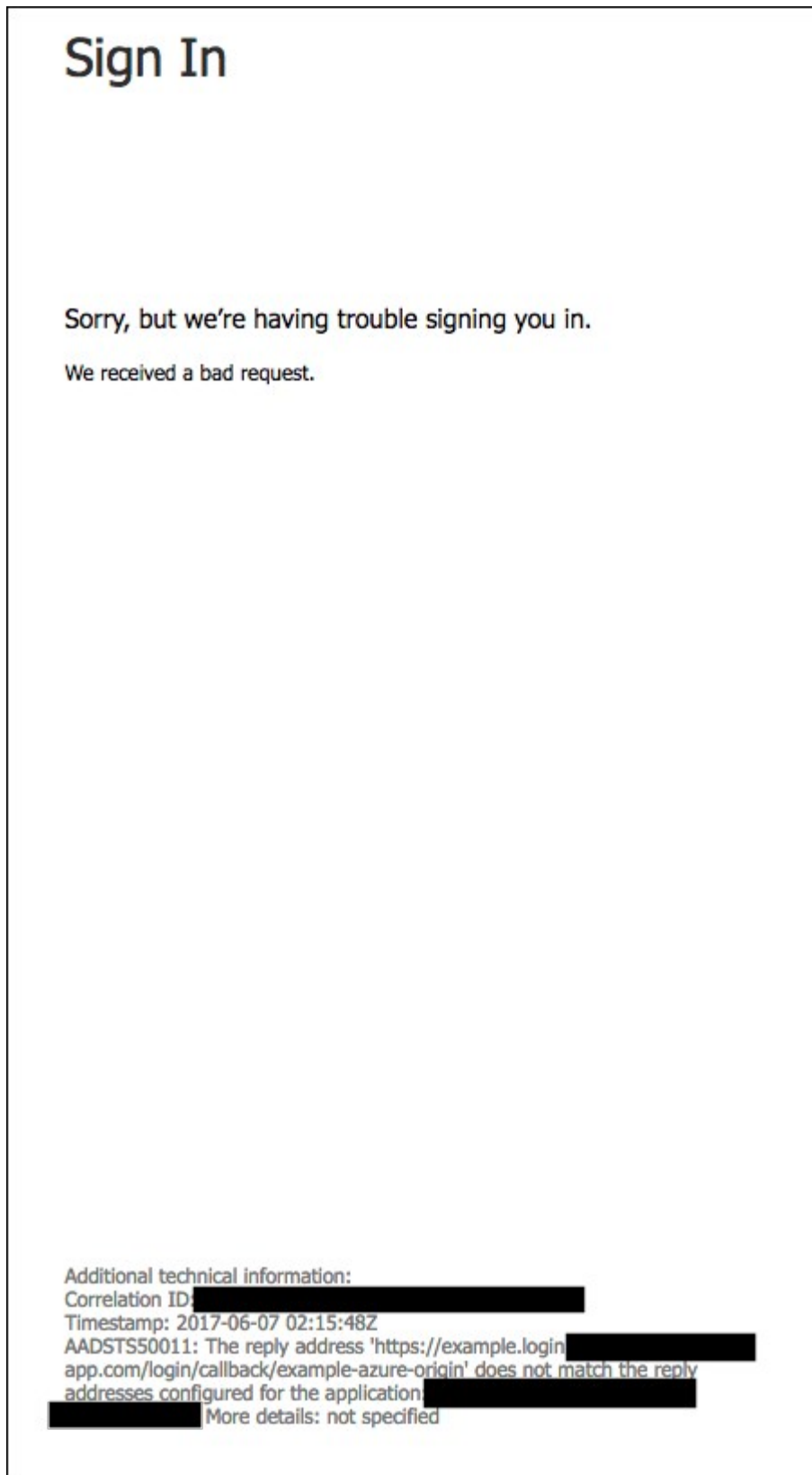
There was an error when authenticating against the external identity provider: Cannot determine username from credentials supplied

Explanation:

- No value is mapped to the username used by Pivotal Platform. 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 might 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:



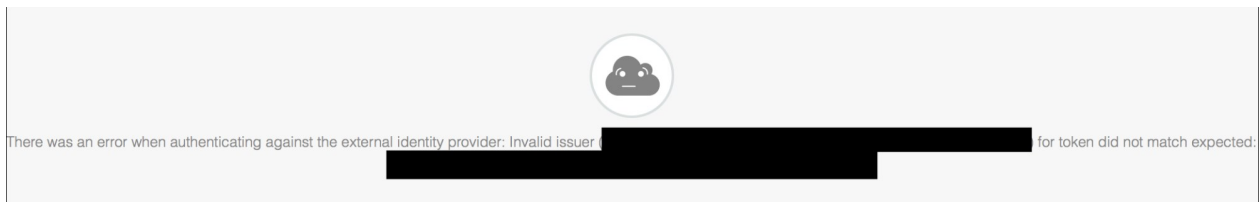
There was an error when authenticating against the external identity provider: 404 Not Found

Explanation:

- The Token Key URL might 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:

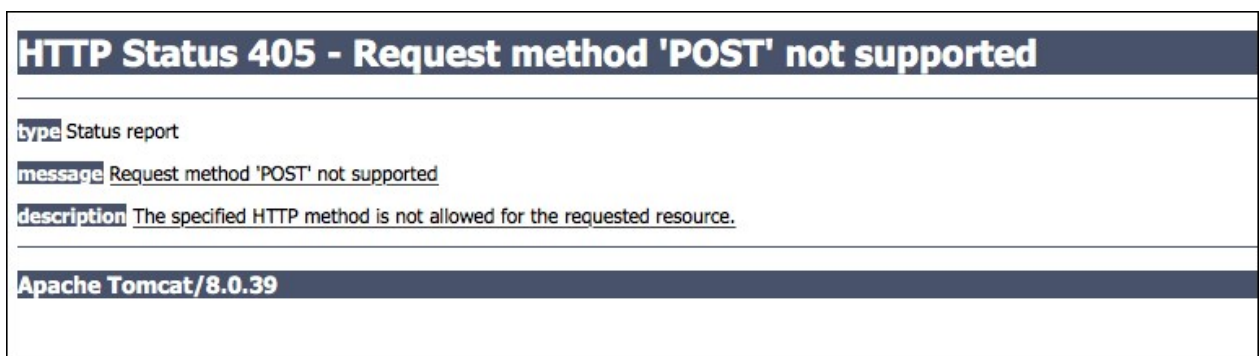


Explanation:

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

Request Method 'POST' not supported (405 Error)

Symptom:

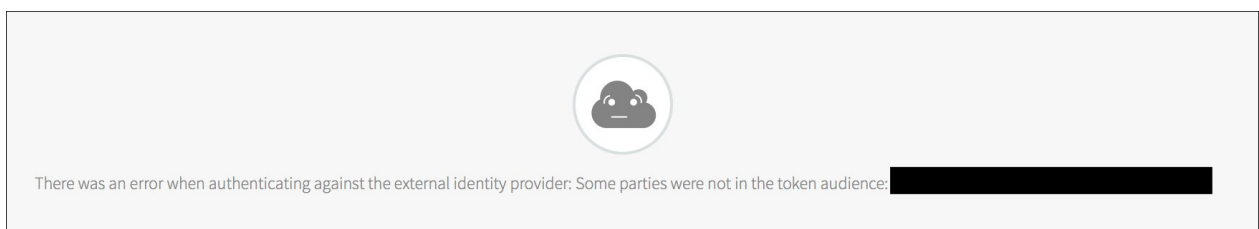


Explanation:

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

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

Symptom:



Explanation:

- The Relying Party Client ID might 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](#)

Layer7 SiteMinder Integration Guide Overview



Warning: Pivotal Single Sign-On v1.11 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.

Layer7 SiteMinder (formally known as CA Single Sign-On) 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 Layer7 SiteMinder as the identity provider and the Pivotal Single Sign-On as the service provider.

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

Prerequisites

To integrate Layer7 SiteMinder with Single Sign-On you must have the following:

- Layer7 SiteMinder v12.52 or later
- A certificate signed 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.

Layer7 SiteMinder Integration Guide

Configuring Layer7 SiteMinder with Single Sign-On

Complete both steps below to integrate your deployment with Layer7 SiteMinder and Single Sign-On .

1. [Configure Layer7 SiteMinder 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 Layer7 SiteMinder as an Identity Provider



Warning: Pivotal Single Sign-On v1.11 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 Layer7 SiteMinder as your identity provider by configuring SAML integration in both Pivotal Platform and Layer7 SiteMinder.

Set up SAML in Pivotal Single Sign- On

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

Plans 1 [New Plan](#)

Name	Sign In Header
CA SSO PCF SSO ▾	example
Edit Plan Manage Identity Providers	

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

Plans > CA SSO PCF SSO > Identity Providers

Identity Providers 2 [New Identity Provider](#) [Configure SAML Service Provider](#)

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

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

Configure SAML Service Provider

[Download Metadata](#)
☒ Perform signed authentication requests

☐ Require signed assertions

[Cancel](#)
[Save](#)

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 Layer7 SiteMinder

1. Sign in as a Layer7 SiteMinder 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 Layer7 SiteMinder 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.

Entities

View Federation Entities • View Entity Return to View Federation Entities

Entity Type

Entity Location: Local
Entity Type: SAML2 IDP

Entity Details

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

Default Signature and Encryption Options

Signing Private Key Alias: signingcert
Signed Authentication Requests Required: No

Supported Name ID Formats and 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.

The screenshot shows the 'Entities' configuration page. The top section is 'Entity Type' with 'Entity Location: Remote' and 'Entity Type: SAML2 SP'. Below is 'Entity Details' with 'Entity ID: http://sso.login.coral.springapps.io', 'Entity Name: pcf-coral', and 'Description: pcf-coral imported via metadata'. The 'Remote Assertion Consumer Service URLs' table has two rows: one for HTTP-POST and one for HTTP-Artifact, both pointing to the same URL. The 'Remote SLO Service URLs' table has two rows: one for HTTP-POST and one for HTTP-redirect, both pointing to the same URL. The 'Manage Name ID Service URLs' table has two rows: one for Binding and one for Location URL, both pointing to the same URL. The 'Signature and Encryption Options' section shows 'Verification Certificate Alias: pcf-coral', 'Encryption Certificate Alias: pcf-coral', and 'Sign Authentication Requests: Yes'. The 'Name ID Formats' section shows 'Supported Name ID Formats' and 'Selected Formats'.

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

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	

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 app 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.

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.

Update session for ForceAuthn

Idle Timeout: 1 : 0 (Hours:Minutes)

Maximum Timeout: 2 : 0 (Hours:Minutes)

Enable Enhanced Session Assurance ☐

Authentication Request Binding: ☒ HTTP-Redirect ☐ HTTP-POST

SSO Binding: ☒ HTTP-Post ☐ HTTP-POST

Audience:

Accept ACS URL in the Authrequest ☐

Transactions Allowed: ☒ Both IDP and SP initiated ☐

SSO Validity Duration (Seconds): 60

Recommended SP Session Duration: ☒ Use Assertion Validity ☐ Customize

Enable Negative Authentication Response ☐

Enable User Consent ☐

User Consent Service URL:

User Consent Post Form:

Minimum Authentication Level: 5

Custom Post Form:

Set 'OneTimeUse' Condition ☐

Validation Period: 0 : 0 : 0 (Hours:Minutes:Seconds)

x	Binding	URL	Default
1	HTTP-POST	https://sso.login.coral.springapps.io/saml/SSO/alias/sso.login.coral.springapps.i	<input checked="" type="checkbox"/>

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.

Partnerships

View Federation Partnerships • Modify Partnership Siteminder/UCP

1 Configure Partnership 2 Federation Users 3 Assertion Configuration 4 SSO and SLO 5 Signature and Encryption

Signature

Disable Signature Processing ☐

Signing Private Key Alias:

Signing Algorithm:

Verification Certificate Alias:

Artifact Signature Options:

Post Signature Options:

☒ Require Signed Authentication Requests

☐ Require Signed ArtifactResolve

☐ Sign ArtifactResponse

Encryption

Encryption Options: ☒ Encrypt Name ID ☐ Encrypt Assertion

Encryption Certificate Alias:

Block Algorithm:

Key Algorithm:

Decryption Private Key Alias:

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

Federation Partnership List							Create Partnership
							1-6 of
Actions	Name	Local Type	Local Entity ID	Remote Type	Remote Entity ID	Status	FIPS Statu
Action	pcf-coral- sso	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	✖

22. 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

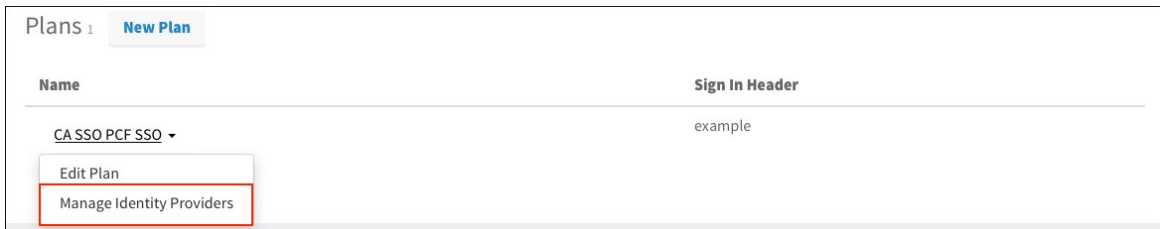


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Set up SAML

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



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

New Identity Provider Cancel Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

<https://idp.company.com/SAML2>

Fetch Metadata

▶ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

Advanced 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)**, 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 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and Layer7 SiteMinder. An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).


Test Your Service Provider Connection

1. Log in to Apps Manager at <https://apps.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 plan bound to your app. Select the service instance 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... >

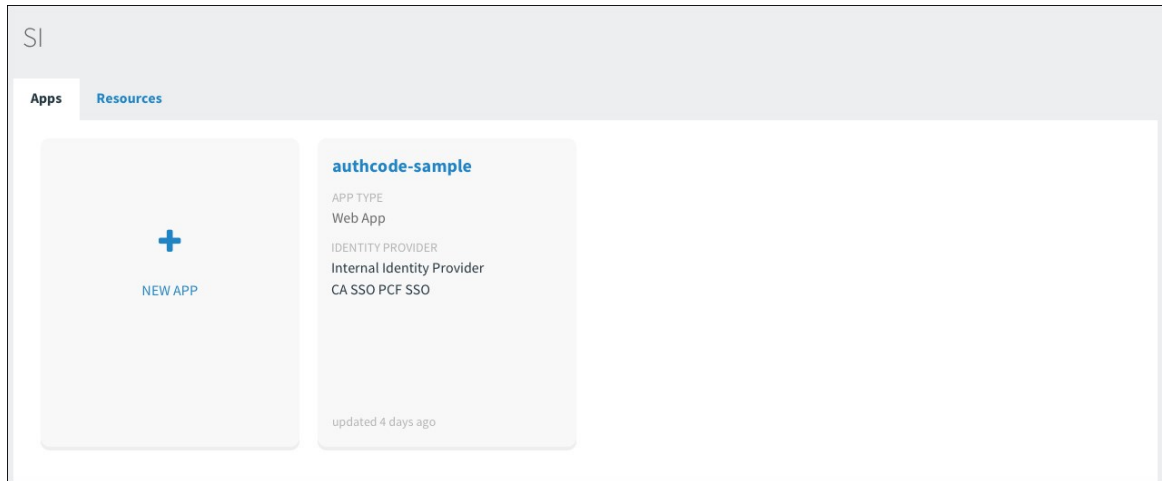
Services

Add Service

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



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



4. Under **Identity Providers**, select the Layer7 SiteMinder 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**

- Return to Apps Manager and click on the URL below your app 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?

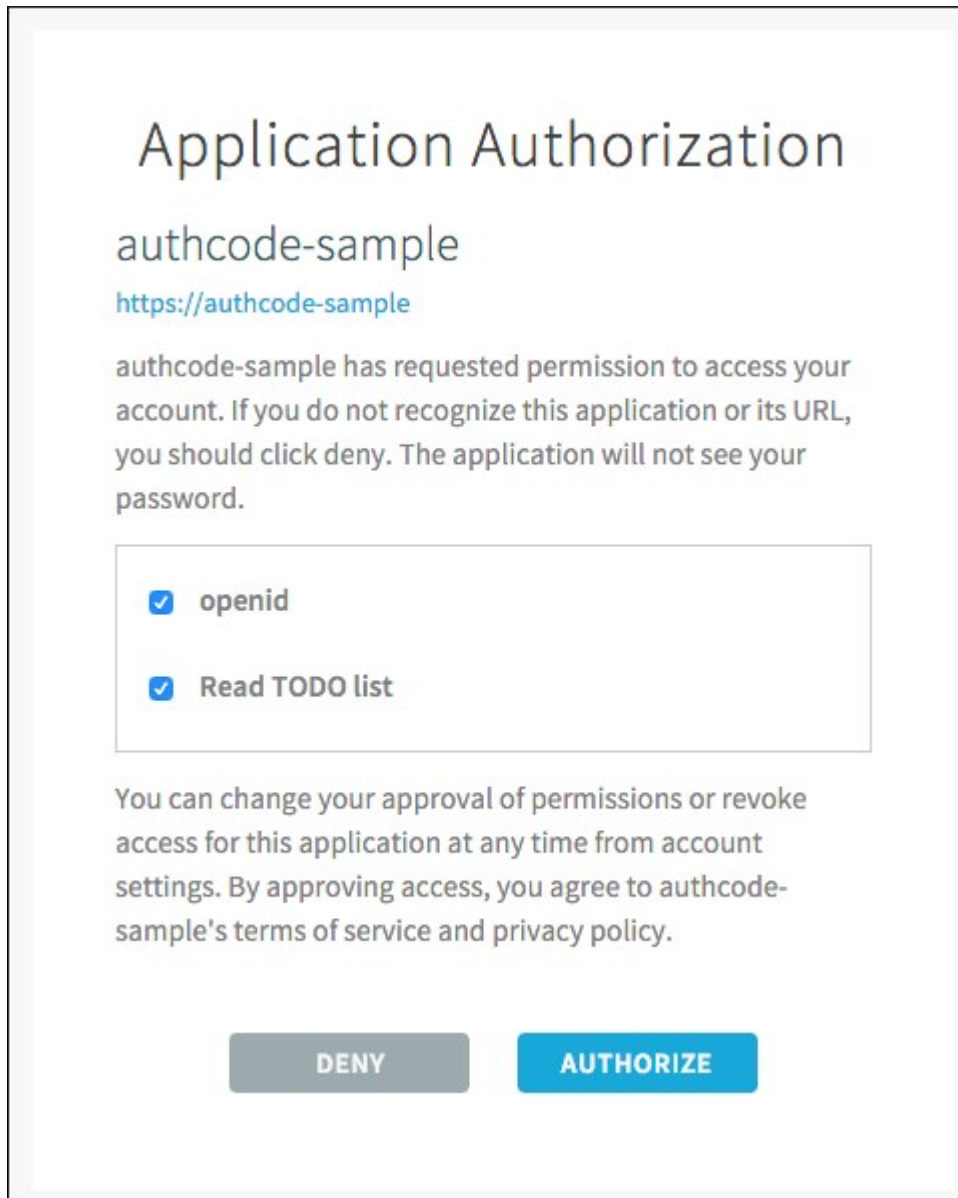
- [Log in via Auth Code Grant Type](#)

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



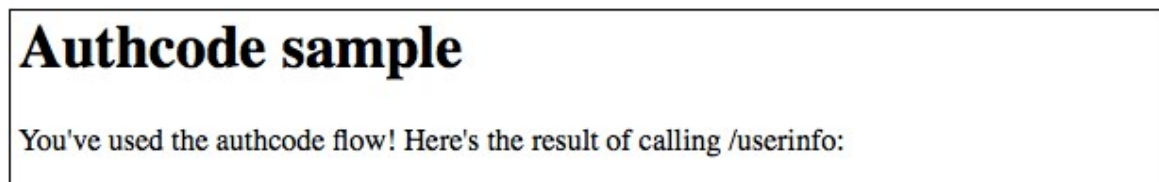
A login form titled "Please Login" with a light orange background. It contains two input fields: "Username:" and "Password:", each followed by a text box with a small eye icon to toggle visibility. Below the fields is a "Login" button.

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



An "Application Authorization" screen for "authcode-sample" with the URL "https://authcode-sample". It informs the user that the application has requested permission to access their account and provides instructions on what to do if the application is unrecognized. Below this, there is a list of requested permissions: "openid" and "Read TODO list", both of which are checked. At the bottom, there are two buttons: "DENY" (grey) and "AUTHORIZE" (blue).

9. The access token and ID token displays.



A section titled "Authcode sample" with the text "You've used the authcode flow! Here's the result of calling /userinfo:".

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

This is the Access Token that was used:

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

This is the ID Token:

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

What do you want to do?

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

Test Your Identity Provider Connection

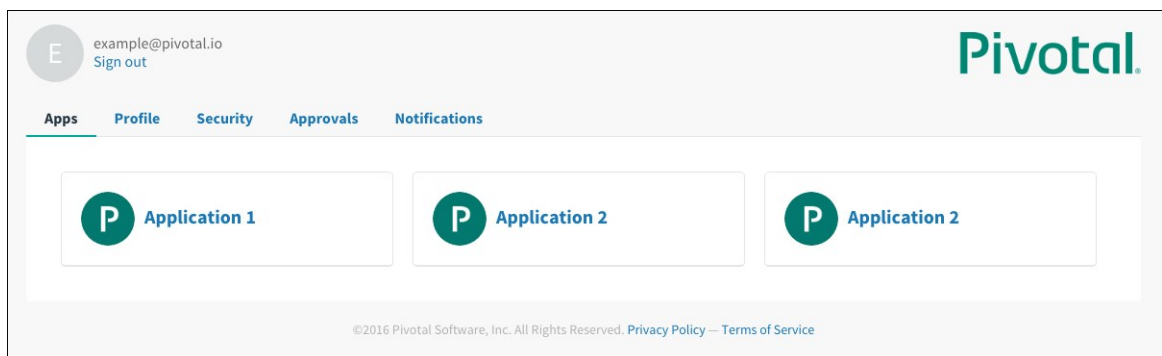


Note: Single Sign-On does not support identity provider-initiated flow into apps, but it does redirect the user to the User Account and Authentication (UAA) page to select apps assigned to the user.

1. Sign in to Layer7 SiteMinder.

The image shows a login form titled "Please Login". It has two input fields: "Username:" and "Password:". Below the password field is a "Login" button. The form is set against a light orange background.

2. Navigate to your app and click it.
3. You are redirected to the page that lists apps 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 Layer7 SiteMinder as well.

1. Sign in to the sample app. 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 Layer7 SiteMinder login page.

Please Login

Username:

Password:

Login

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

Troubleshooting



Warning: Pivotal Single Sign-On v1.11 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 PLayer7 SiteMinder and Pivotal Single Sign- On.

Layer7 SiteMinder Partnership is Inactive

Symptom:

The following error occurred: 403 - Request Forbidden. Transaction ID: d5dd24a-950bf795-1a3cf7c7-0bcb12dc-82689d7c-bc failed.

Explanations:

- The Layer7 SiteMinder is inactive in Layer7 SiteMinder.

Service Provider Entity ID Misconfigured

Symptom:

HTTP Status 403 - Request Forbidden. Transaction ID: 174f32c9-98739353-1c861a37-2f05277b-847a8663-988 failed.

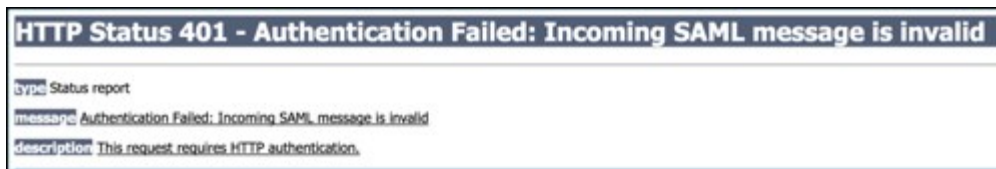
type	Status report
message	Request Forbidden. Transaction ID: 174f32c9-98739353-1c861a37-2f05277b-847a8663-988 failed.
description	Access to the specified resource has been forbidden.

Explanation:

- The service provider Entity ID is misconfigured in Layer7 SiteMinder.

Incoming SAML message is invalid

Symptom:

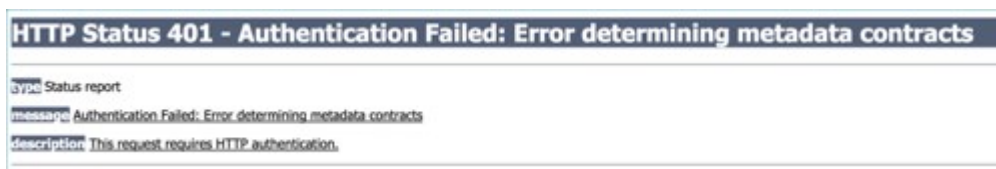


Explanation:

- The identity provider Entity ID is misconfigured in Layer7 SiteMinder or in Single Sign- On.
- The Name ID Format was misconfigured in Layer7 SiteMinder.

Assertion Consumer Service URL Misconfigured

Symptom:

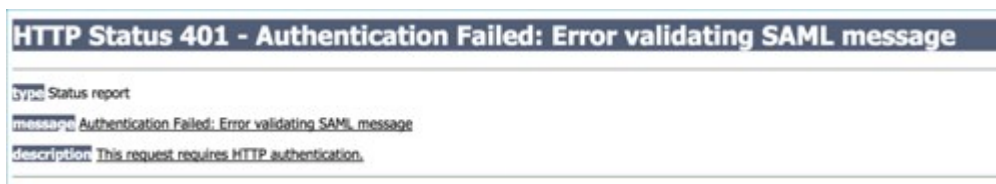


Explanation:

- The service provider Assertion Consumer Service (ACS) is misconfigured in Layer7 SiteMinder.

Audience Field Misconfigured

Symptom:



Explanation:

- The service provider Audience Field is misconfigured in Layer7 SiteMinder.

Expired Certificate

Symptom:



Explanation:

- The certificate has expired in Layer7 SiteMinder.

Identity Provider SSO URL Misconfigured

Symptom:



Explanation:

- The identity provider SSO URL is misconfigured in 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 Overview



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On to use Google Cloud Platform (GCP) as an OpenID Connect (OIDC) identity provider.

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

Prerequisites

To integrate GCP as a single sign-on identity provider for Pivotal Platform apps, you must have the following:

- A Single Sign-On service plan with **Plan Administrators** and **Organizations** configured. See [Manage Service Plans](#).
- An active Google Cloud project.
- A GCP user account with project editor or higher privileges.

Integrate Google Cloud Platform OIDC for Single Sign-On

Complete the step below to set up GCP as an OIDC identity provider for Single Sign-On.

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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On service plan by configuring OpenID Connect (OIDC) integration in both Single Sign-On and GCP.

Overview

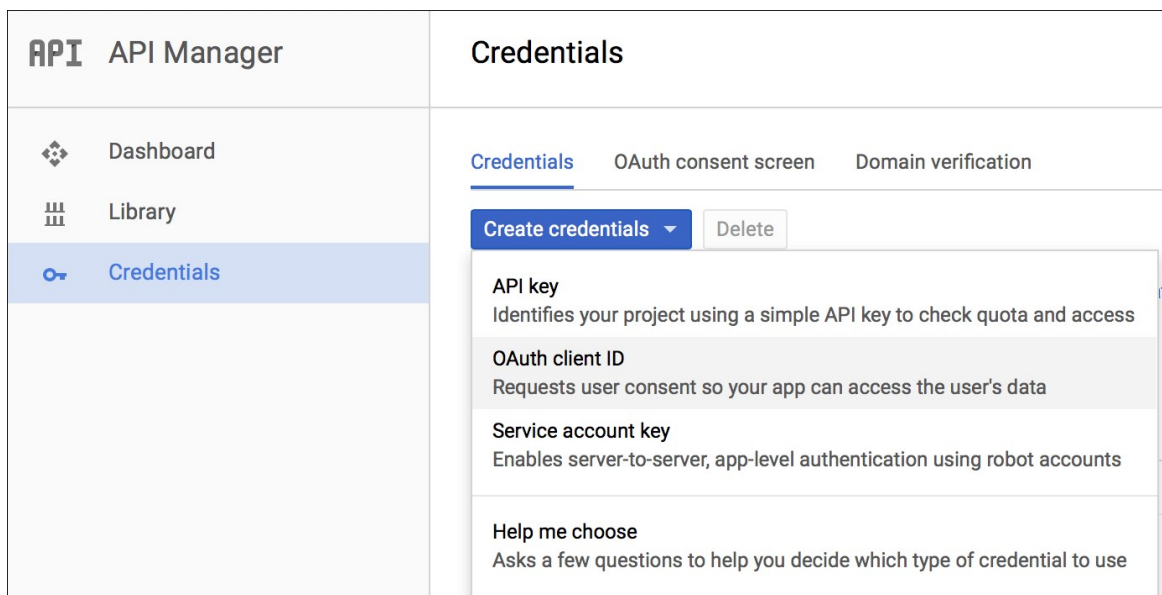
To set up the integration, follow the procedures below:

1. [Generate GCP Client Credentials](#)
2. [Set up the OIDC Identity Provider in Single Sign-On](#)

Generate GCP Client Credentials

Follow the steps below to generate GCP client credentials:

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



- 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 using the following pattern:

```
https://AUTH-DOMAIN.login.SYSTEM-DOMAIN/login/callback/ORIGIN-KEY
```

Where:

- AUTH-DOMAIN** is the **Auth Domain** you entered in [Create or Edit Service Plans](#).
- ORIGIN-KEY** is identical to the **Identity Provider Name** you set later in the SSO Operator Dashboard in [Set Up OIDC Identity Provider in Single Sign-On](#), except that it cannot include uppercase letters or spaces.



Warning: The origin key does not change after it is assigned, even if the **Identity Provider Name** is modified.

Application type

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

Name

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

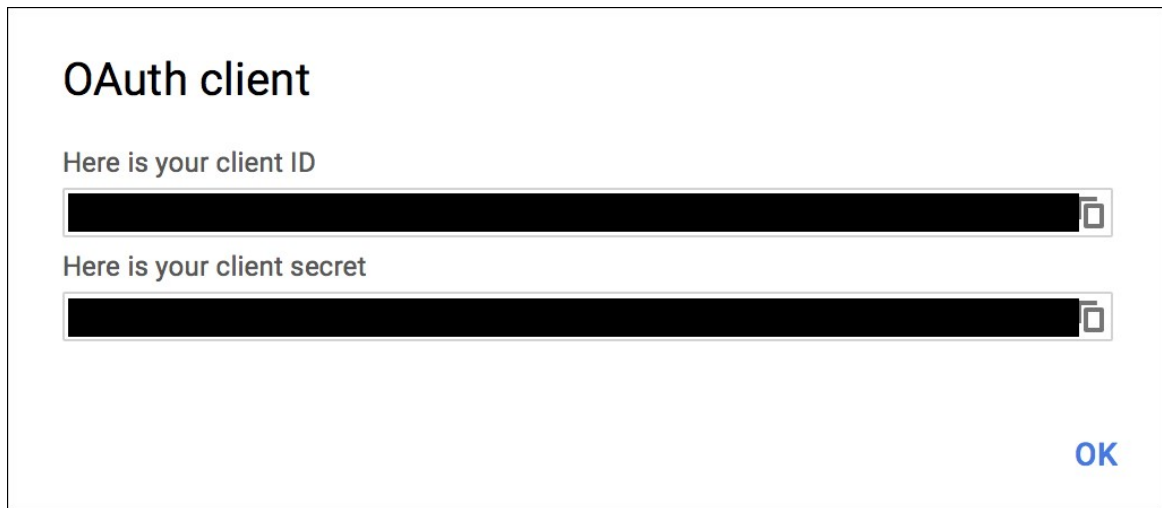
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (http://*.example.com) or a path (http://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

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.

×

- Click **Create** and record the **client ID** and **client secret** generated. You enter these values as your **Relying Party OAuth Client ID** and **Relying Party OAuth Client Secret** in the SSO Operator Dashboard in [Set Up OIDC Identity Provider in Single Sign-On](#) below.



OAuth client

Here is your client ID

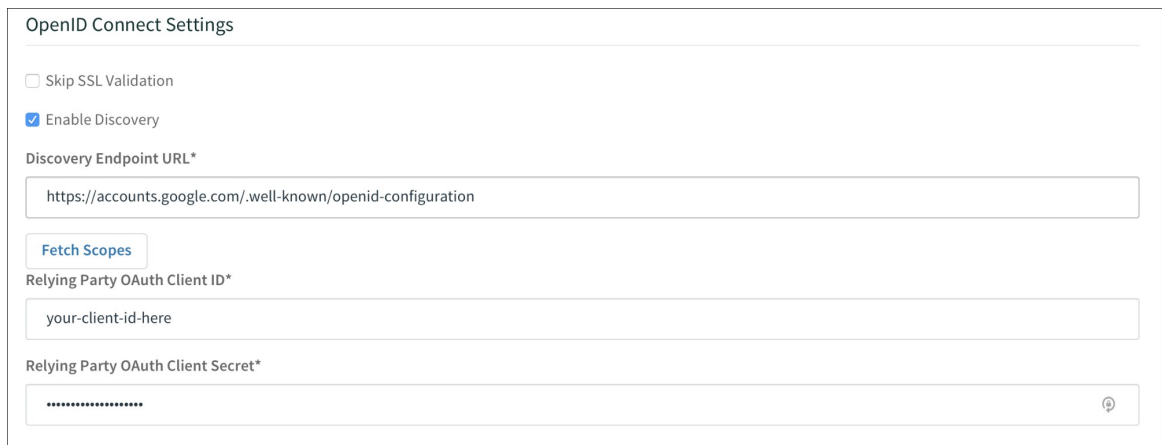
Here is your client secret

OK

Set up the OIDC Identity Provider in Single Sign- On

Follow the steps below to set up the OIDC identity provider in Single Sign- On:

1. Follow steps 1 – 6 in [Add an OIDC Provider](#).
2. In the **Discovery Endpoint URL** field, enter `https://accounts.google.com/.well-known/openid-configuration`.
3. Click **Fetch Scopes**.
4. 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](#)

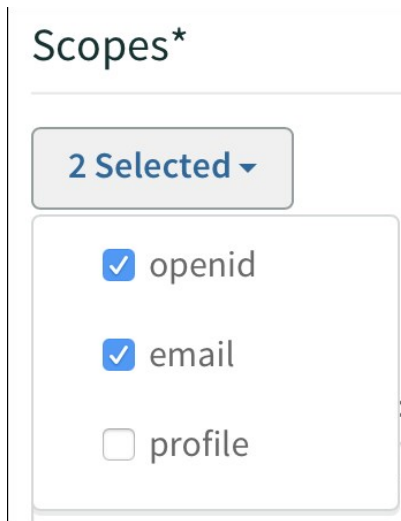
Relying Party OAuth Client ID*

`your-client-id-here`

Relying Party OAuth Client Secret*

.....

5. Ensure that `openid` and `email` are selected as scopes. You can select additional scopes if you want.

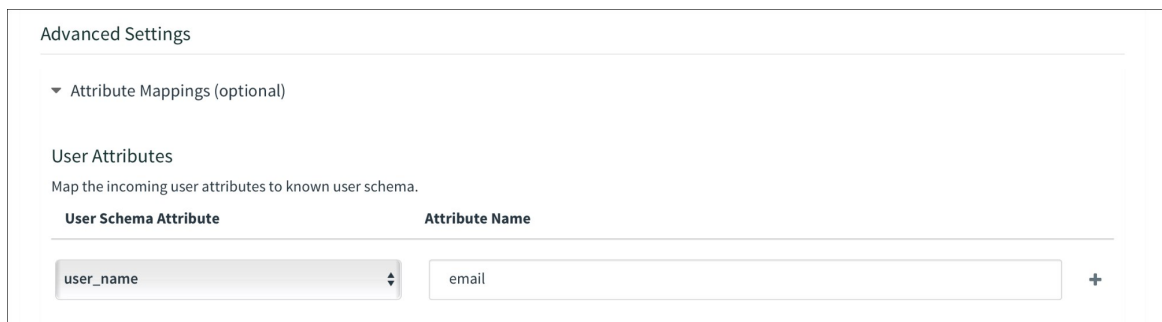


Scopes*

2 Selected ▾

- ☒ openid
- ☒ email
- ☐ profile

6. Under **Advanced Settings** > **Attribute Mappings (optional)** > **User Attributes**, select **email** as the **User Schema Attribute** and enter `user_name` as the **Attribute Name**. This enables Single Sign-On to identify the authenticated user.



Advanced Settings

▼ Attribute Mappings (optional)

User Attributes

Map the incoming user attributes to known user schema.

User Schema Attribute	Attribute Name
user_name	email

+

7. (Optional) Configure additional attribute mappings.
8. Click **Create Identity Provider** to save your settings.
9. (Optional) [Enable IdP Discovery](#) for the service plan.

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

Testing



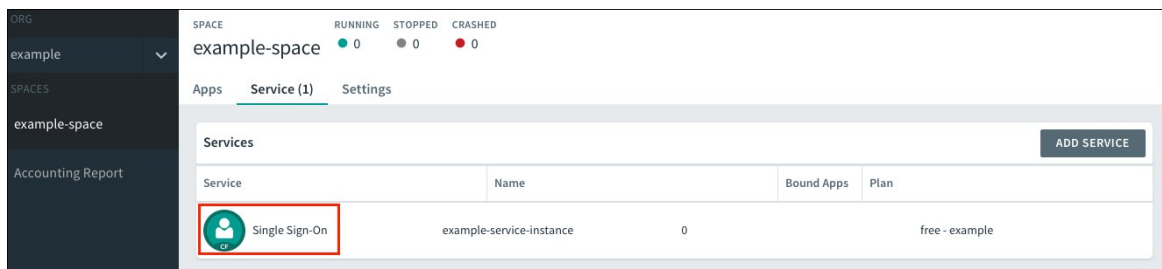
Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how Pivotal Platform administrators can test the OpenID Connect (OIDC) connection between Pivotal Single Sign-On and Google Cloud Platform.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

Test Your Single Sign-On Connection

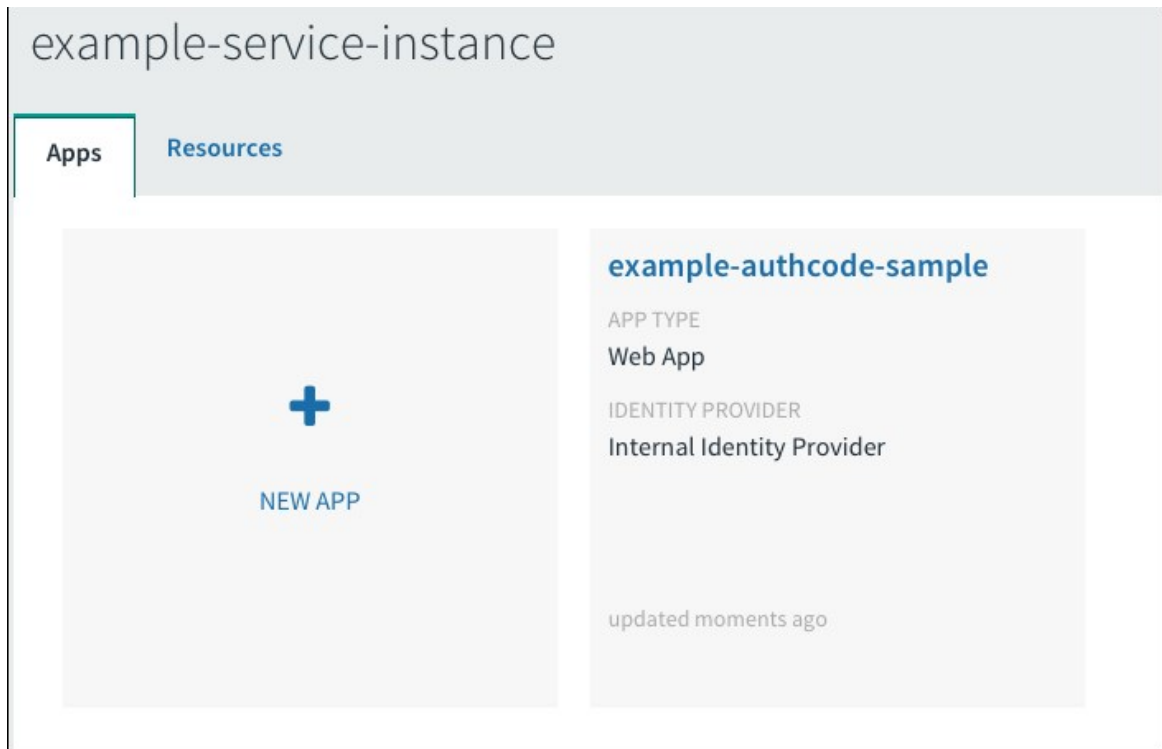
1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN> and navigate to the org and space where your app is located.
2. Under **Services**, locate the service instance of the Single Sign-On plan bound to your app.



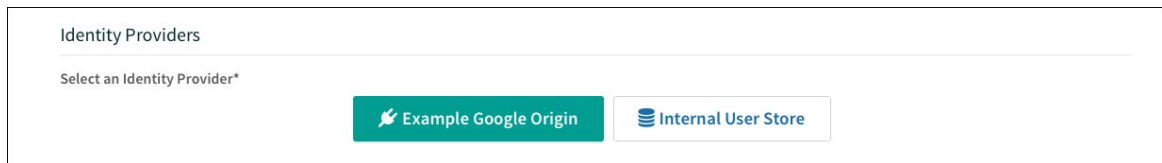
3. Select the service instance and click **Manage**.



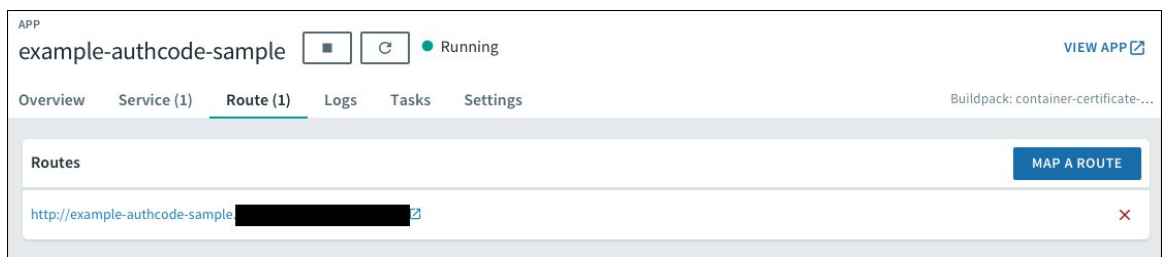
4. Under the **Apps** tab, select your app.



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



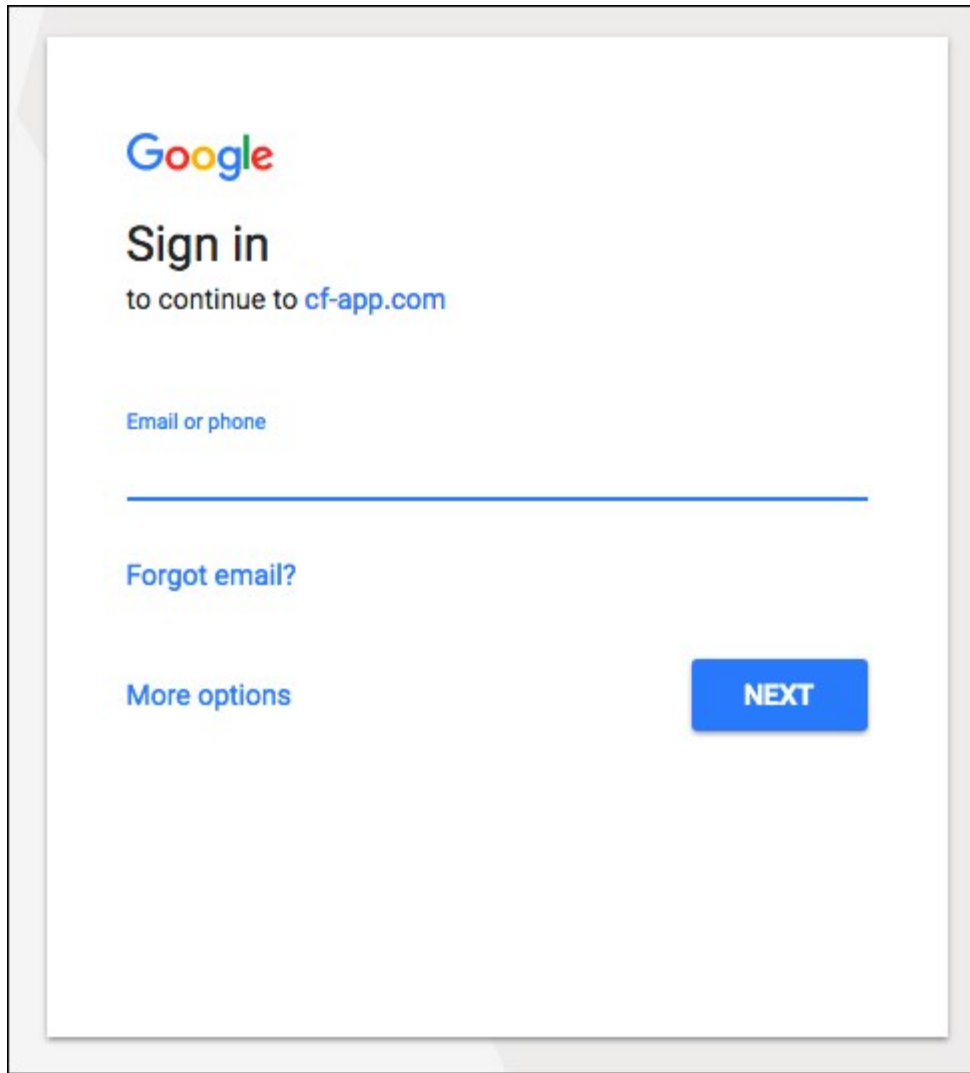
- Return to Apps Manager and click the URL listed below your app to access your app.



- 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.



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 Single Sign- On 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: Pivotal Single Sign-On v1.11 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.

No Link for OIDC

Symptom:

Welcome to Example!

Email

Password

SIGN IN

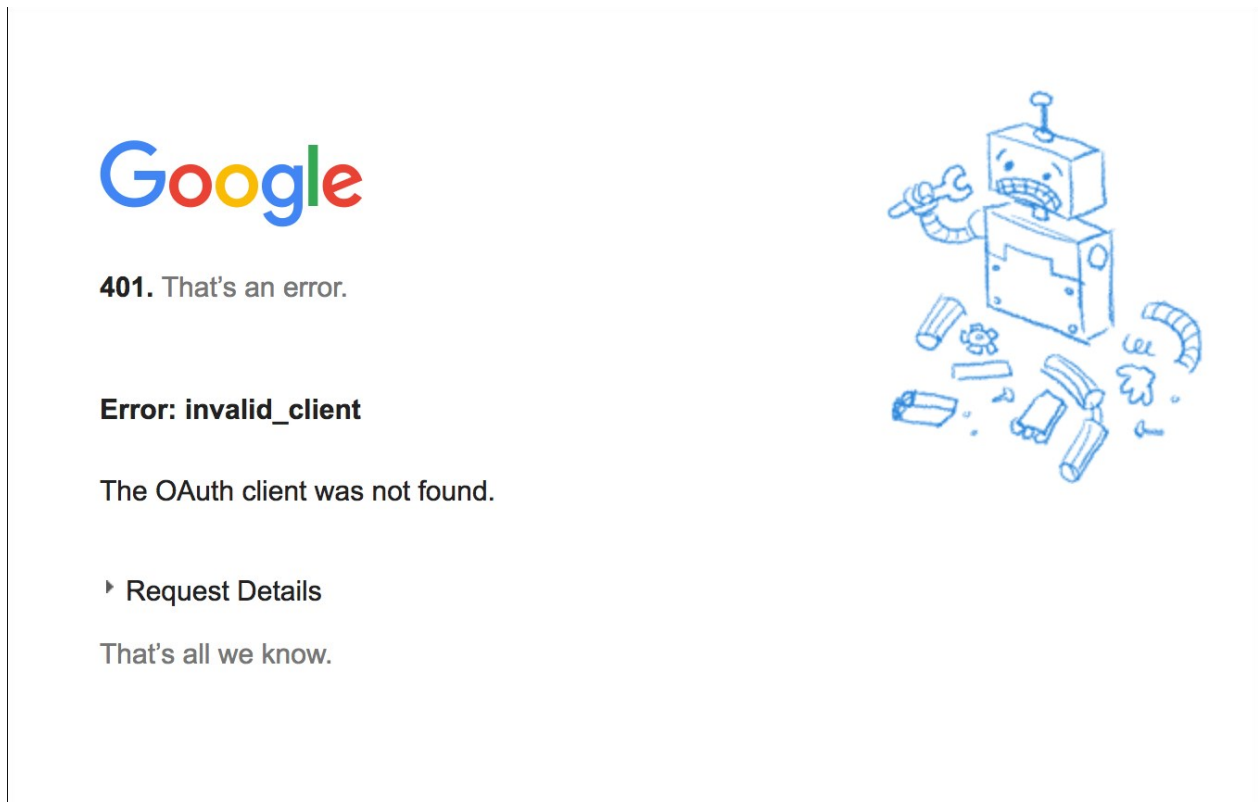
Create account Reset password

Explanation:

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

No OAuth Client Found

Symptom:

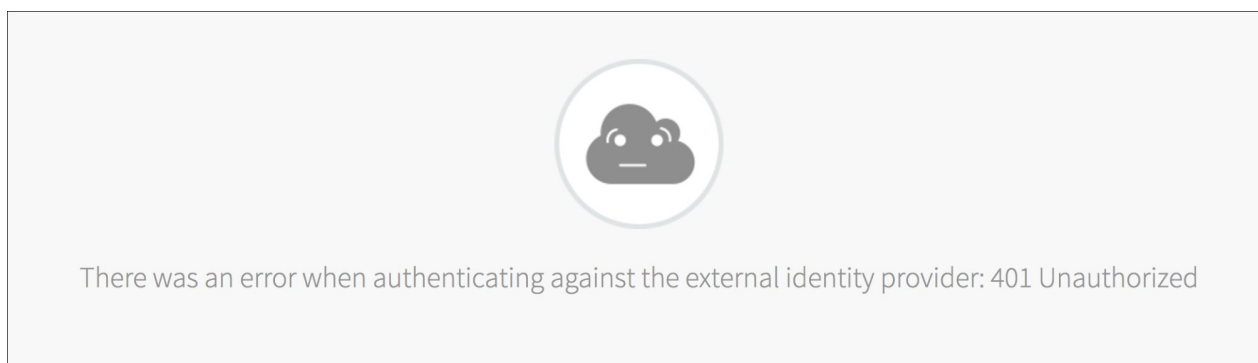


Explanation:

- Incorrect OAuth Client ID configured.

Unauthorized

Symptom:




Explanation:

- Incorrect OAuth client secret configured.

Redirect URI Mismatch

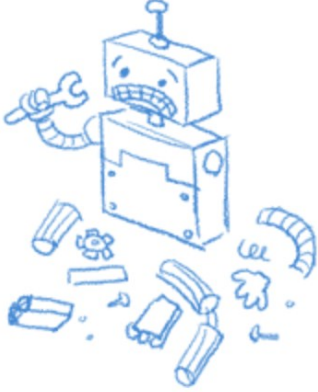
Symptom:



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.

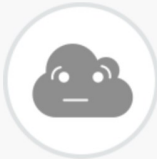


Explanation:

- Incorrect authorization redirect URI on OAuth Client.

Empty Username

Symptom:



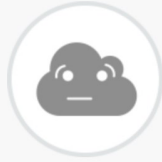
There was an error when authenticating against the external identity provider: Username cannot be empty

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 Overview



Warning: Pivotal Single Sign-On v1.11 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 apps 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 and Pivotal Single Sign-On as the service provider.

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

Prerequisites

To integrate Okta with Single Sign-On, you must have the following:

- Okta v2016.07 or later
- A user with app 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 Single Sign-On

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

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: Pivotal Single Sign-On v1.11 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.

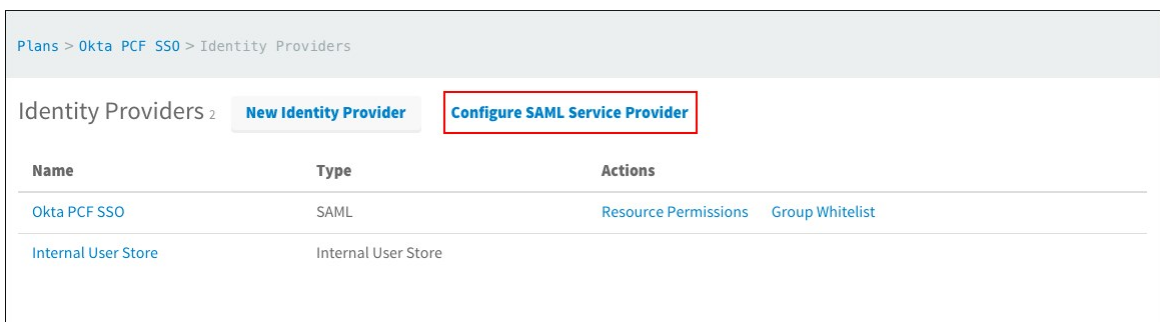
This topic describes how to set up Okta as your identity provider by configuring SAML integration in both Pivotal Single Sign-On and Okta.

Set up SAML in Single Sign-On

1. Log into the SSO Operator 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 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**.



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

Configure SAML Service Provider

[Download Metadata](#)☒ Perform signed authentication requests☐ Require signed assertions

Cancel

Save

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

Set Up SAML in Okta

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

The screenshot shows the Okta PCF SSO configuration page. The 'Sign On' tab is selected. Under the 'Settings' section, the 'SAML 2.0' option is highlighted with a red box. Below this, a message states 'SAML 2.0 is not configured until you complete the setup instructions.' with a 'View Setup Instructions' button. The 'CREDENTIALS DETAILS' section shows 'Application username format' set to 'Okta username' and 'Password reveal' set to 'Allow users to securely see their password (Recommended)'. The right sidebar contains 'About' and 'Application Username' sections.

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:

- Enter the **AssertionConsumerService Location URL** from your downloaded service provider metadata into **Single sign on URL**.
For example, `https://PORTAL-FQDN/saml/SSO/alias/PORTAL-FQDN`.
Where **PORTAL-FQDN** is the fully qualified domain name (FQDN) for your login portal. The portal FQDN uses the format `AUTH-DOMAIN.login.SYSTEM-DOMAIN`. You can view the portal FQDN for a plan by logging into the SSO Operator Dashboard, clicking the name of your plan, and selecting **Edit Plan**.
- Enter the FQDN for your login portal into **Audience URI (SP Entity ID)**. This value is available in the downloaded service provider metadata as the entity ID.
- Select a **Name ID format**.
- 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](#).

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 'Sign On' tab is active, showing a 'Settings' section with an 'Edit' button. The 'Settings' section has a 'SIGN ON METHODS' subsection. It shows 'SAML 2.0' as the selected method. Below this, there's a 'Default Relay State' field. A yellow warning box states: 'SAML 2.0 is not configured until you complete the setup instructions.' with a 'View Setup Instructions' button. Below the warning box, a red box highlights the text 'Identity Provider metadata' which is followed by 'is available if this application supports dynamic configuration.' To the right of the settings, there's an 'About' section explaining SAML 2.0 and an 'Application Username' section with instructions on how to choose a format and a note about manual entry if 'None' is selected. At the bottom, there's a 'CREDENTIALS DETAILS' section with fields for 'Application username format' (set to 'Okta username') and 'Password reveal' (with a checkbox for 'Allow users to securely see their password (Recommended)').

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

Configuring a Single Sign-On Service Provider

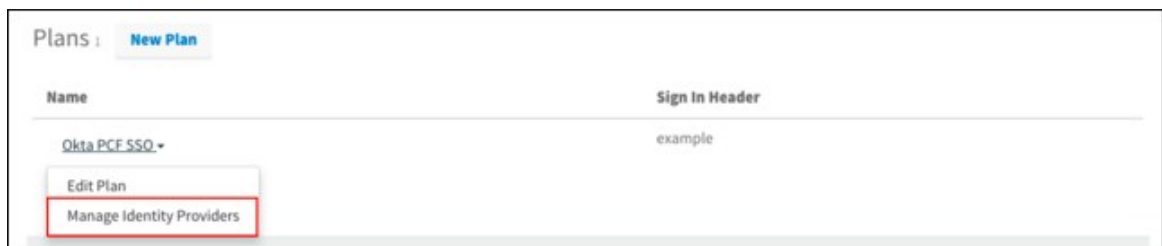


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Setting up SAML

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



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

New Identity Provider Cancel Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

<https://idp.company.com/SAML2>

Fetch Metadata

▶ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

Advanced 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and Okta services. An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

Test Your Service Provider Connection

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

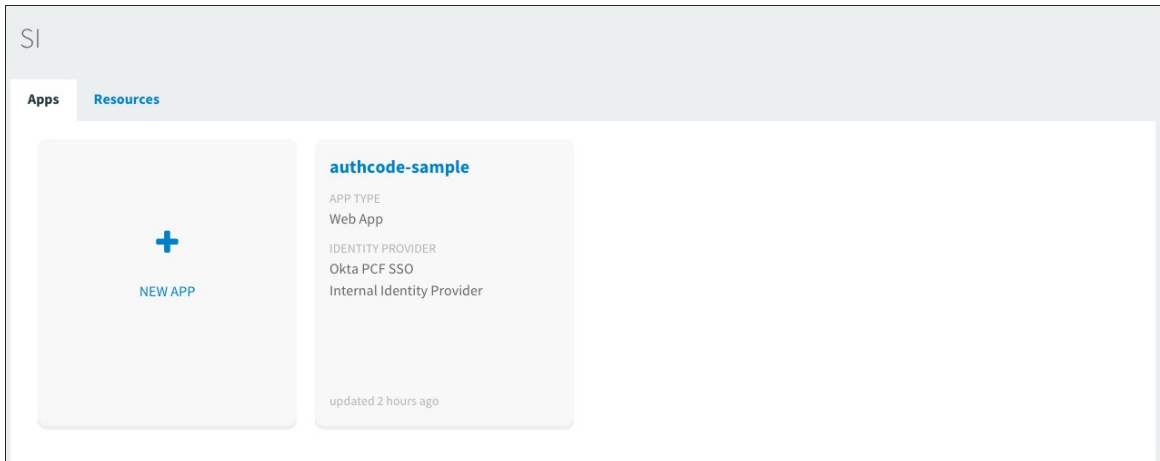
The screenshot shows the 'Services' section of the Apps Manager. It contains a table with the following data:

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

A red box highlights the 'Manage' button next to the 'Pivotal Single Sign-On' service instance.

The screenshot shows the details page for the 'Pivotal Single Sign-On' service instance. The 'Manage' button is highlighted with a red box. The page also shows the 'App Binding (1)' section with the 'authcode-sample' application listed.

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



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

The screenshot shows the configuration page for the 'authcode-sample' app. The 'App Name' is 'authcode-sample'. Under 'Identity Providers', 'Okta PCF SSO' is selected and highlighted with a red box. The 'Redirect URIs' section shows 'https://authcode-sample.id-service.cf-app.com'. The 'Authorization' section shows 'Scopes' with 'todo.read' and 'todo.write' selected, and 'System Provided' with 'openid' selected. At the bottom, there are 'Delete', 'Cancel', and 'Save Config' buttons.

- Return to Apps Manager and click on the URL below your app 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**.

A screenshot of the Pivotal sign-in page. At the top left is the Pivotal logo. The main heading is 'Sign In'. There are 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 section titled 'Your security image' with a large question mark icon. At the bottom left, there are links for 'Forgot password?' and 'Help'.

8. The app 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

- 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", "openid", "27b2d43c-2f0d-48e8-979c-b11f841e972d" ],
"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?

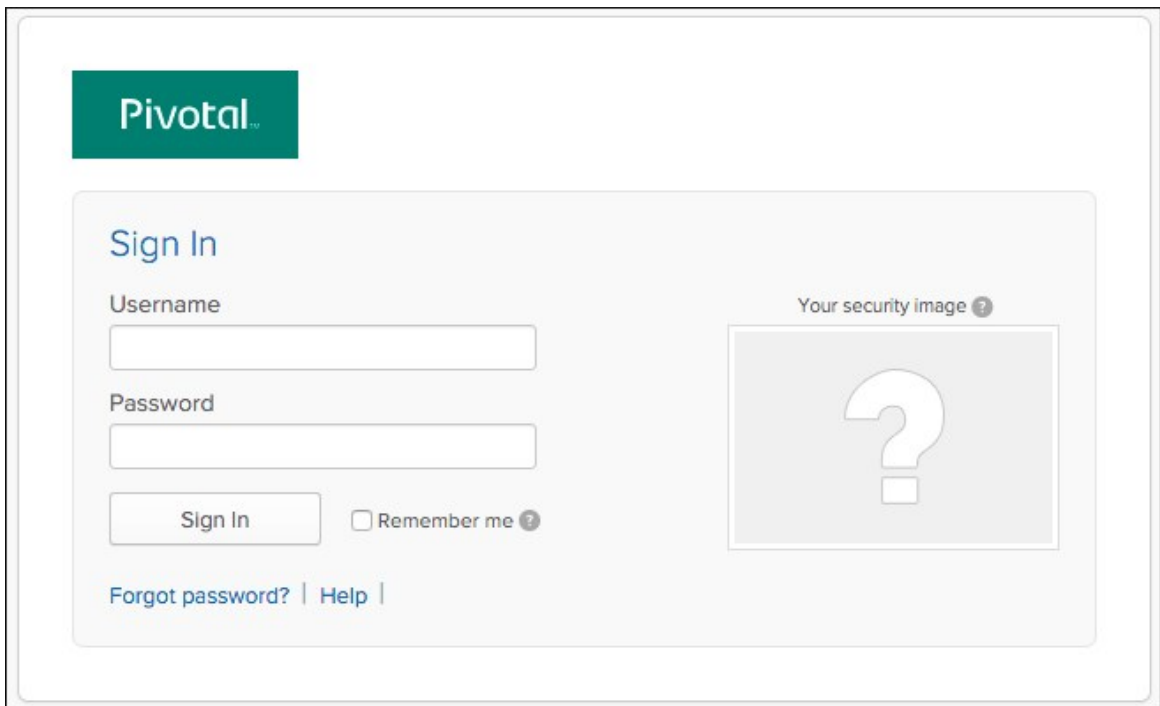
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



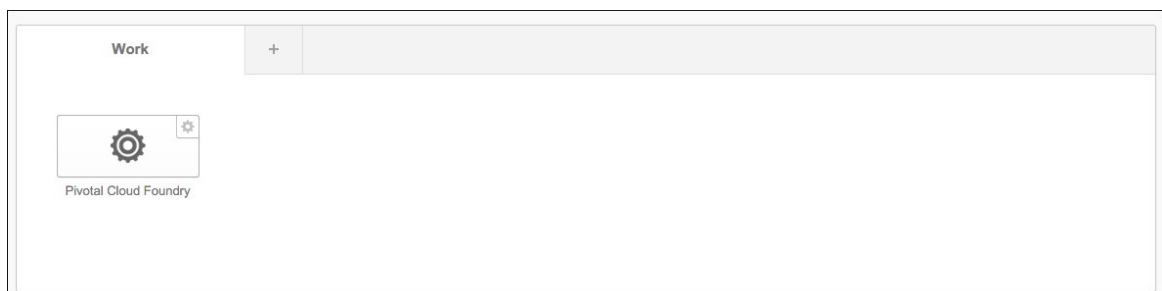
Note: Single Sign-On does not support identity provider-initiated flow into apps, but it does redirect the user to the User Account and Authentication (UAA) page to select apps assigned to the user.

1. Sign into Okta.

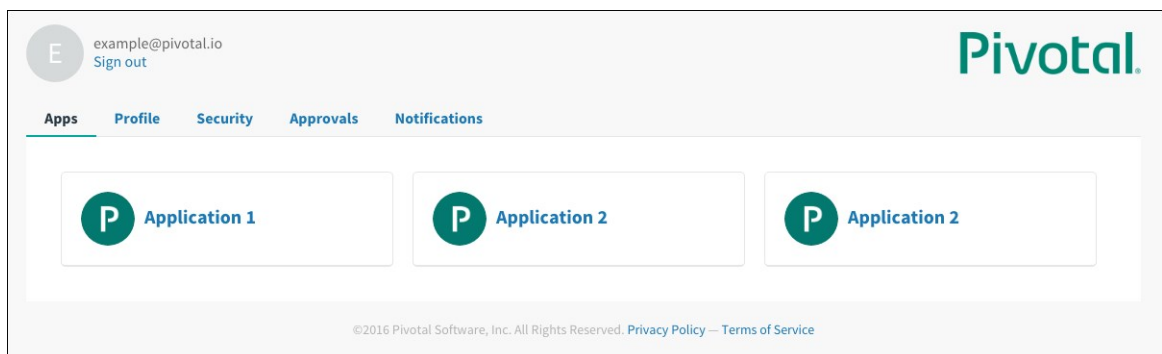


The image shows the Pivotal Sign In page. At the top left is the Pivotal logo. Below it is a 'Sign In' section with two input fields: 'Username' and 'Password'. Below the password field is a 'Sign In' button and a 'Remember me' checkbox with a help icon. To the right of the sign-in fields is a placeholder 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'.

2. Navigate to the app tile and click it.



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



Test Your Single Sign-Off

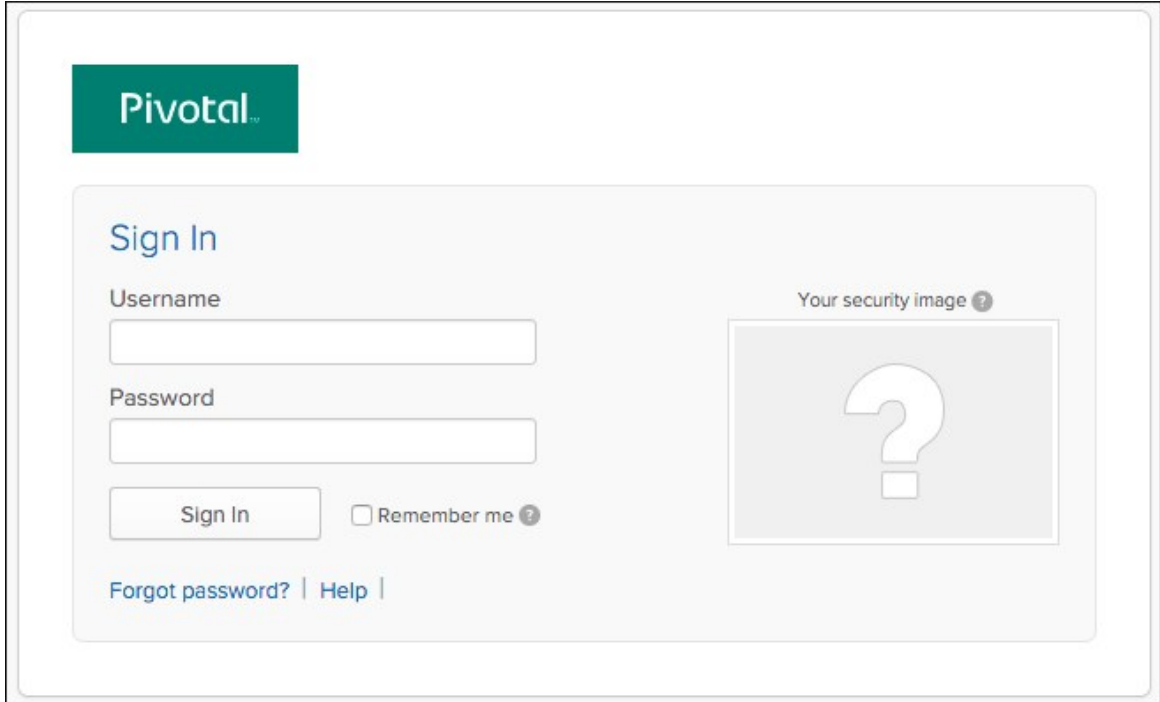
Test single sign-off to ensure that when users log out of the app, they are logged out of Okta as well.

1. Sign into the sample app. 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 Okta login page.

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

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

Troubleshooting

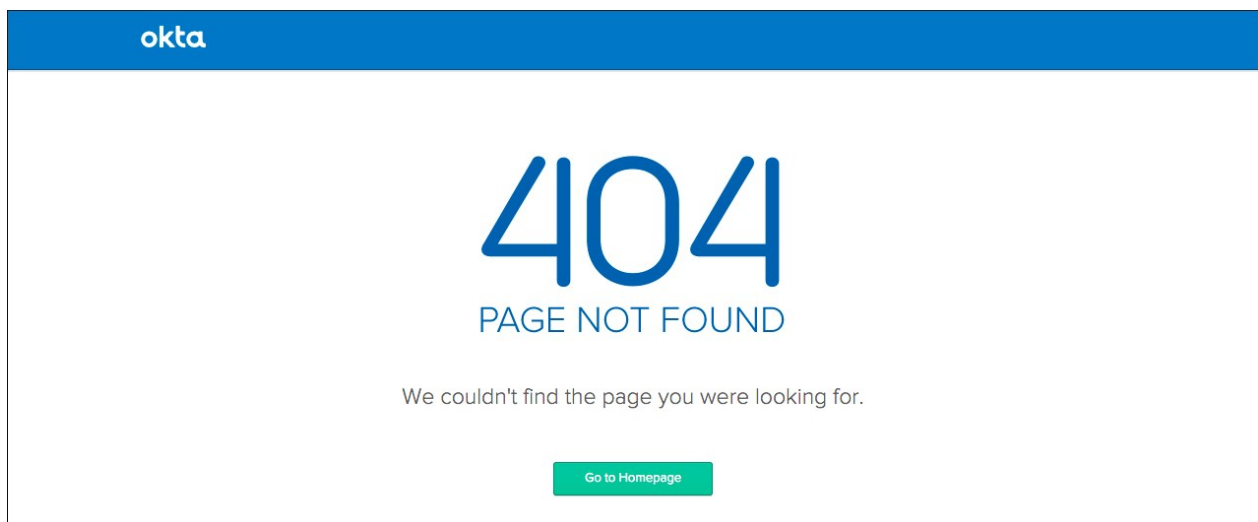


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

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 Single Sign-On plan settings.

No Valid Assertion

Symptom:



Response doesn't have any valid assertion which would pass subject validation

Explanations:

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

Webpage Not Available

Symptom:



This webpage is not available

DNS_PROBE_FINISHED_NXDOMAIN

[Details](#)

Explanation:

- The SSO URL is misconfigured in Okta.

Metadata Not Found

Symptom:



Metadata for issuer <http://www.okta.com/exk5s2s8y0ugC73JY0h7> wasn't found

Explanation:

- The identity provider Entity ID is misconfigured in the Single Sign- On plan settings.

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

PingFederate Integration Guide Overview



Warning: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On as the Service Provider (SP).

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

Prerequisites

To integrate PingFederate with Single Sign-On, you must have the following:

- 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 Single Sign-On

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

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

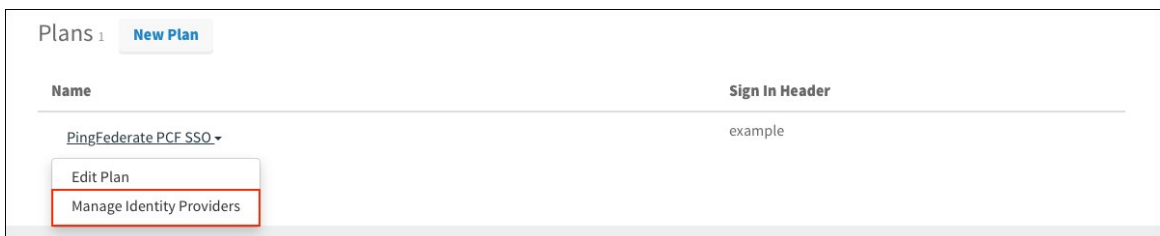


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

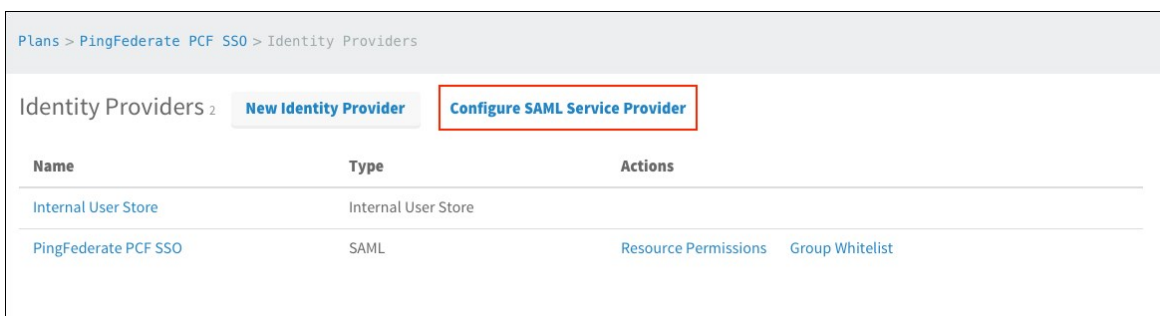
This topic describes how to set up PingFederate as your identity provider by configuring SAML integration in both Pivotal Single Sign-On and PingFederate.

Set up SAML in Single Sign-On

1. Log into the SSO Operator Dashboard at <https://p-identity.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.

Configure SAML Service Provider

[Download Metadata](#)

- ☒ Perform signed authentication requests
- ☐ Require signed assertions

Cancel

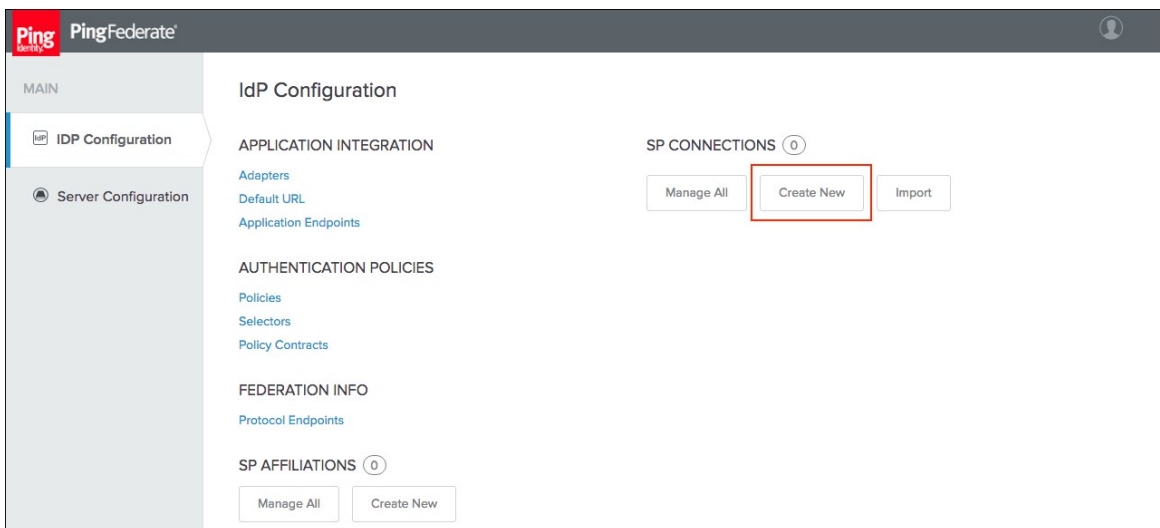
Save

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

Set up SAML in PingFederate

Configure the Connection

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



- Select the **Browser SSO Profiles** connection template on the **Connection Type** tab and click **Next**.
- Select **Browser SSO** on the **Connection Options** tab and click **Next**.
- Select **File** as the method for importing metadata and click **Choose file** to choose the SSO metadata on the **Import Metadata** tab. Click **Next**.

PingFederate

MAIN

- IDP Configuration
- Server Configuration

SP Connection

Connection Type | Connection Options | **Import Metadata** | General Info | Browser SSO | Credentials | Activation & Summary

To populate many connection settings automatically, you can upload the partner's metadata file, or specify a URL where PingFederate can download it. To periodically reload the connection settings from the URL, select Enable Automatic Reloading.

METADATA: ☐ NONE ☒ **FILE** ☐ URL

spring_saml_metadata.xml Choose file

Cancel Previous **Next**

- Review the information on the **Metadata Summary** tab and click **Next**.
- Ensure that the **Partner's Entity ID**, **Connection Name**, and **Base URL** fields pre-populate based on the metadata. Click **Next**.

PingFederate

MAIN

- IDP Configuration
- Server Configuration

SP Connection

Connection Type | Connection Options | Import Metadata | **Metadata Summary** | General Info | Browser SSO | Credentials

Activation & Summary

This information identifies your partner's unique connection identifier (Connection ID). Connection Name represents the plain-language identifier for this connection. Optionally, you can specify multiple virtual server IDs for your own server to use when communicating with this partner. If set, these virtual server IDs will be used in place of the unique protocol identifier configured for your server in Server Settings. The Base URL may be used to simplify configuration of partner endpoints.

PARTNER'S ENTITY ID (CONNECTION ID): example.login.id-serv

CONNECTION NAME: example.login.id-serv

VIRTUAL SERVER IDS: Add

BASE URL: https://example.login.id-service.cf-app.cc

Configure Browser SSO

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

PingFederate

MAIN

- IDP Configuration
- Server Configuration

SP Connection | Browser SSO

SAML Profiles | Assertion Lifetime | Assertion Creation | Protocol Settings | Summary

A SAML Profile defines what kind of messages may be exchanged between an Identity Provider and a Service Provider, and how the messages are transported (bindings). As an IdP, you configure this information for your SP connection.

Single Sign-On (SSO) Profiles: ☒ **IDP-INITIATED SSO** ☒ **SP-INITIATED SSO**

Single Logout (SLO) Profiles: ☐ IDP-INITIATED SLO ☐ SP-INITIATED SLO

Cancel Save Draft **Next**

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

Assertion Creation

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

The screenshot shows the PingFederate web interface for 'Assertion Creation'. The left sidebar has 'MAIN' with 'IDP Configuration' and 'Server Configuration'. The top navigation bar shows 'SP Connection | Browser SSO | Assertion Creation'. Below this are tabs: 'Identity Mapping', 'Attribute Contract' (selected), 'Authentication Source Mapping', and 'Summary'. A descriptive text states: 'An Attribute Contract is a set of user attributes that this server will send in the assertion.' The main area has two sections: 'Attribute Contract' and 'Subject Name Format'. In the 'Subject Name Format' section, a dropdown menu is open, showing 'urn:oasis:names:tc:SAML:1:nameid-format:unspecified' selected. Below this is the 'Extend the Contract' section with an 'Attribute Name Format' dropdown showing 'urn:oasis:names:tc:SAML:2.0:attrname-format:basic' and an 'Add' button. At the bottom are 'Cancel', 'Save Draft', 'Previous', and 'Next' buttons.

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.

The screenshot shows the PingFederate web interface for 'IdP Adapter Mapping'. The left sidebar is the same. The top navigation bar shows 'SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping'. Below this are tabs: 'Adapter Instance' (selected), 'Mapping Method', 'Attribute Contract Fulfillment', 'Issuance Criteria', and 'Summary'. A descriptive text states: '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.' The main area has an 'ADAPTER INSTANCE' section with a dropdown menu showing 'Adapter' selected. Below this is the 'Adapter Contract' section with a text input field containing 'username'. There is an unchecked checkbox for 'OVERRIDE INSTANCE SETTINGS' and a 'Manage Adapter Instances' button. At the bottom are 'Cancel', 'Save Draft', and 'Next' buttons.

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

PingFederate

MAIN

IDP Configuration

Server Configuration

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

☐ 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
☒ USE ONLY THE ADAPTER CONTRACT VALUES IN THE SAML ASSERTION

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

MAIN

IDP Configuration

Server Configuration

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**

PingFederate

SP Connection | Browser SSO | Protocol Settings

Assertion Consumer Service URL | Allowable SAML Bindings | Artifact Lifetime | Artifact Resolver Locations | Signature Policy

Summary

As the IdP, you send SAML assertions to the SP's Assertion Consumer Service. The SP may request that the SAML assertion be sent to one of several URLs, via different bindings. Please provide the possible assertion consumer URLs below and select one to be the default.

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

☐ - SELECT -

[Cancel](#) [Previous](#)

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

PingFederate

SP Connection | Browser SSO | Protocol Settings

Assertion Consumer Service URL | Allowable SAML Bindings | Artifact Resolver Locations | Signature Policy | Encryption Policy

Summary

When the SP sends messages, what SAML bindings do you want to allow?

☐ ARTIFACT

☒ POST

☐ REDIRECT

☐ SOAP

[Cancel](#) [Previous](#)

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**.

PingFederate

MAIN

IDP Configuration

Server Configuration

SP Connection | Credentials

Digital Signature Settings Summary

You may need to digitally sign SAML messages or security tokens to protect against tampering. Please select a key/certificate to use from the list below.

SIGNING CERTIFICATE 21:51:3D:A7:E1:5F (cn=Pivotal)

☒ INCLUDE THE CERTIFICATE IN THE SIGNATURE <KEYINFO> ELEMENT.

☐ INCLUDE THE RAW KEY IN THE SIGNATURE <KEYVALUE> ELEMENT.

SIGNING ALGORITHM RSA SHA256

Manage Certificates

Cancel Save Draft 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

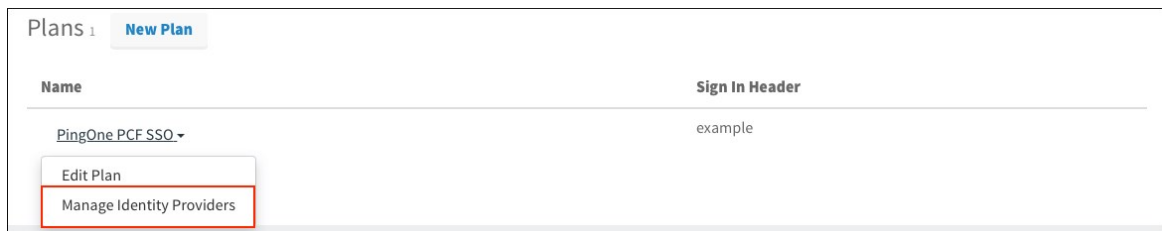


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Set up SAML

1. Log in to the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> as a Plan Administrator.
2. Select your plan and choose **Manage Identity Providers** from the drop-down menu.



3. Click **New Identity Provider**.

New Identity Provider

Cancel **Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

<https://idp.company.com/SAML2>

Fetch Metadata

▶ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

Advanced 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.
9. 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and PingFederate. An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

Test Your Service Provider Connection

1. Log in to Apps Manager at <https://apps.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 plan bound to your app. Click the service instance and then click **Manage**.

The screenshot shows the 'Overview' tab of the Apps Manager. It features two main sections: 'Apps' and 'Services'. The 'Apps' section contains a table with columns: NAME, INSTANCES, MEMORY, LAST PUSH, and ROUTE. One app, 'authcode-sample', is listed with 1 instance, 512MB memory, and a last push of 4 days ago. The 'Services' section contains a table with columns: SERVICE, NAME, BOUND APPS, and PLAN. A service instance named 'Pivotal Single Sign-On' is listed, which is highlighted with a red box. It has 1 bound app and is on the 'free - (MONTHLY)' plan. A red box also highlights the 'Manage' button next to the service icon.

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

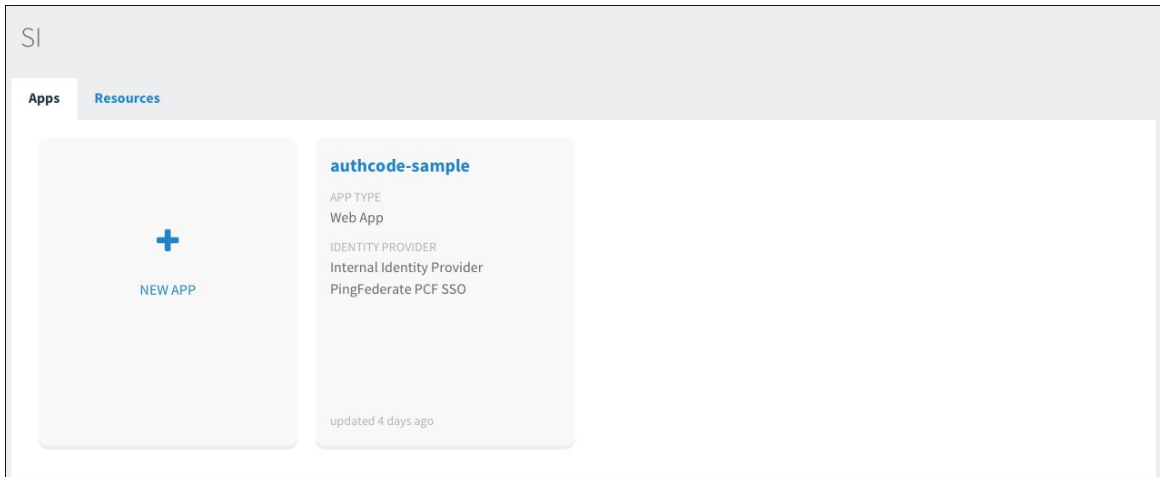
SERVICE	NAME	BOUND APPS	PLAN
	Pivotal Single Sign-On	1	free - (MONTHLY)

The screenshot shows the 'Manage' page for the 'Pivotal Single Sign-On' service instance. The page header includes the service icon, name, instance name 'SI', and service plan 'PingFederate PCF SSO'. Below the header, there are tabs for 'App Binding (1)', 'Plan', and 'Settings'. The 'App Binding (1)' tab is selected, showing a list of bound apps. The 'authcode-sample' app is listed. A red box highlights the 'Manage' button in the top left corner of the page.

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

Bound Apps
authcode-sample

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



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

The screenshot shows the configuration page for the 'authcode-sample' app. It includes fields for 'App Name' (authcode-sample), 'Identity Providers' (with 'Internal User Store' and 'PingFederate PCF SSO' buttons, the latter highlighted with a red box), 'Redirect URIs' (https://authcode-sample.id-service.cf-app.com), 'Authorization' (with 'Scopes' set to 'todo' and 'System Provided' set to 'openid'), and 'Select Scopes' (with 'Auto-Approved Scopes' set to 'None selected'). At the bottom are 'Delete', 'Cancel', and 'Save Config' buttons.

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

Sign On

Username

Password

Login

- The app 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" ],
  ...
}
```



```

"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?

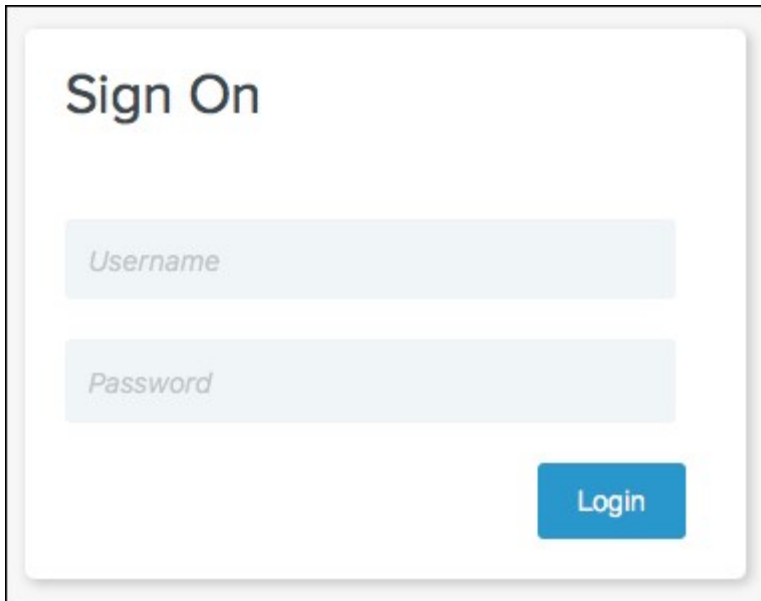
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



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

1. Sign in to PingFederate.



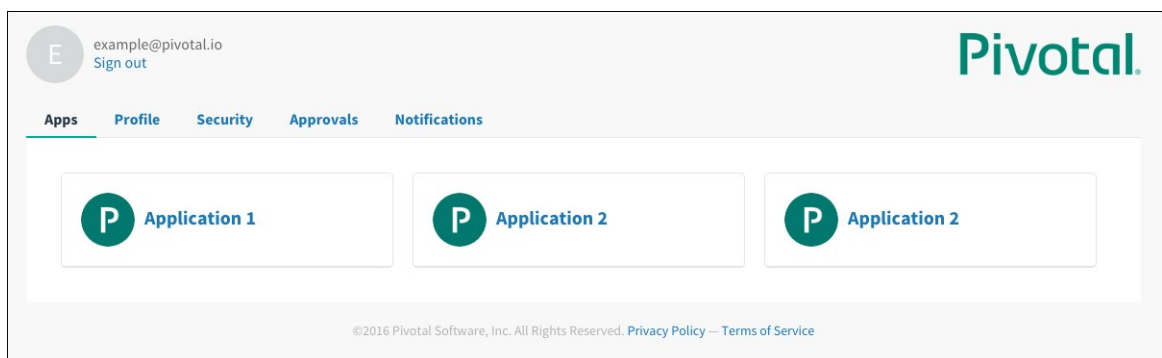
Sign On

Username

Password

Login

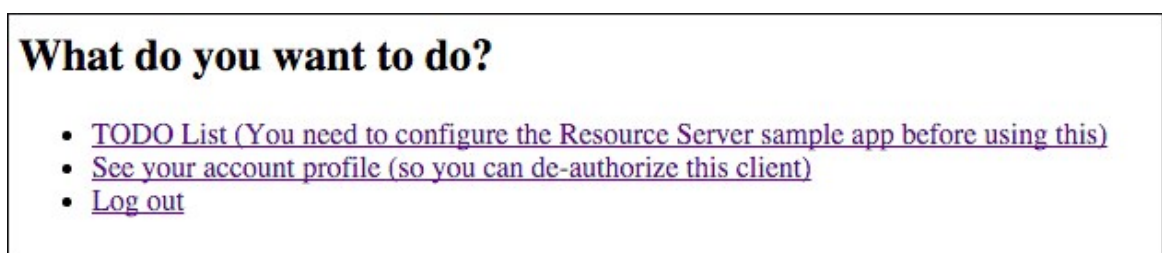
2. Navigate to your app and click it.
3. View the list of apps you have access to.



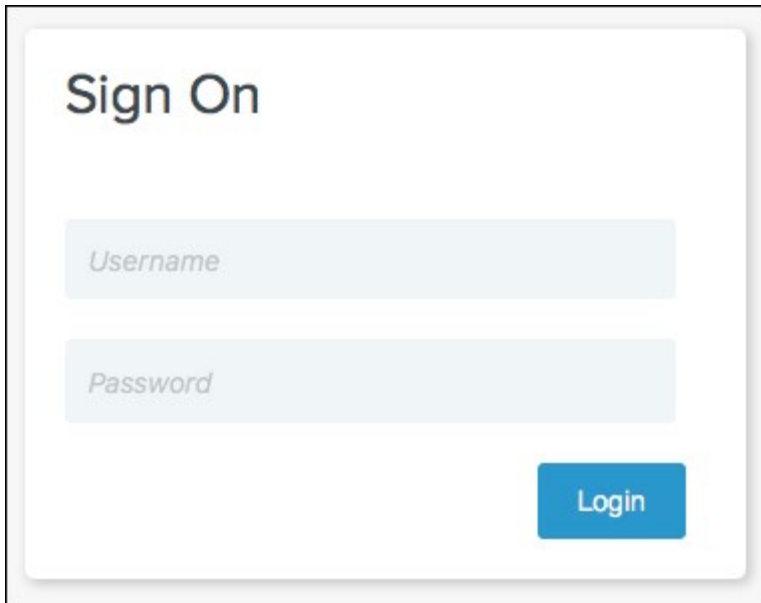
Test Your Single Sign-Off

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

1. Sign into the sample app. 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 is white with a thin black border and a subtle drop shadow. At the top left, the text 'Sign On' is displayed in a large, bold, dark grey font. Below this, there are two light blue input fields. The first field is labeled 'Username' in a smaller, italicized grey font. The second field is labeled 'Password' in the same style. To the right of the 'Password' field, there is a blue rectangular button with the word 'Login' in white text.

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

Troubleshooting

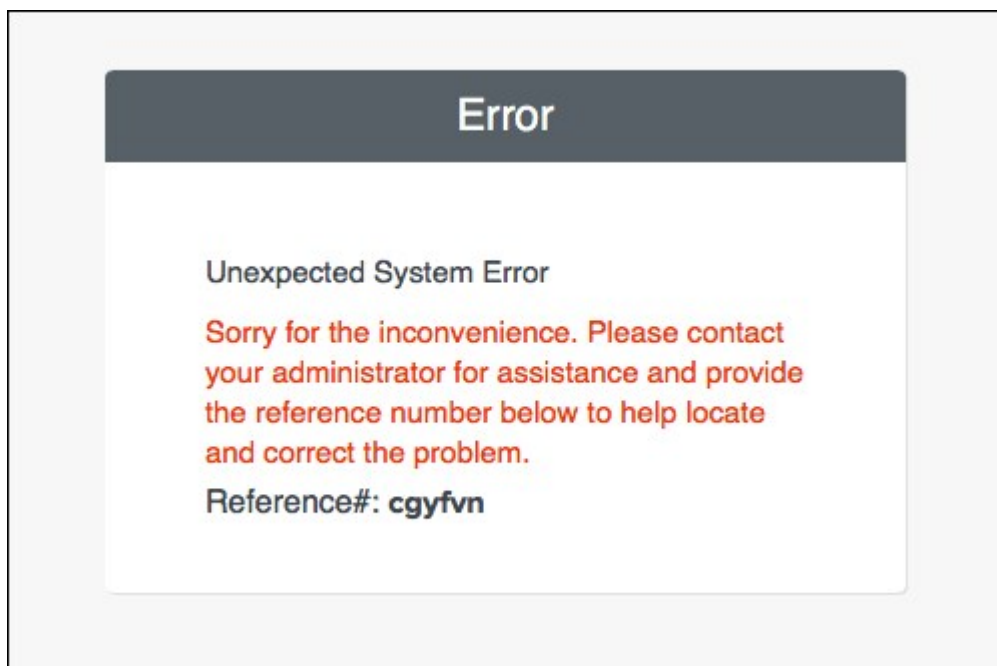


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Error

Symptom:

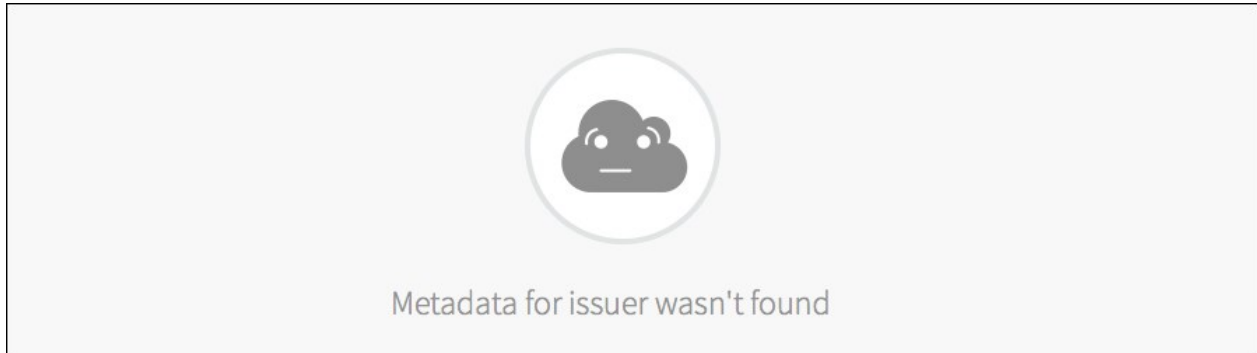


Explanations:

- 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 Single Sign-On plan settings.

Metadata Not Found

Symptom:



Explanation:

- The identity provider Entity ID is misconfigured in the Single Sign- On plan settings.

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

PingOne Cloud Integration Guide Overview



Warning: Pivotal Single Sign-On v1.11 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 apps. This documentation describes how to configure a single sign-on partnership between PingOne Cloud as the Identity Provider (IdP) and Pivotal Single Sign-On as the Service Provider (SP).

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

Prerequisites

To integrate PingOne Cloud with Single Sign-On, you must have the following:

- PingOne Cloud
- A user with app 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 Single Sign-On

Complete both steps below to integrate your deployment with PingOne Cloud and Single Sign-On.

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

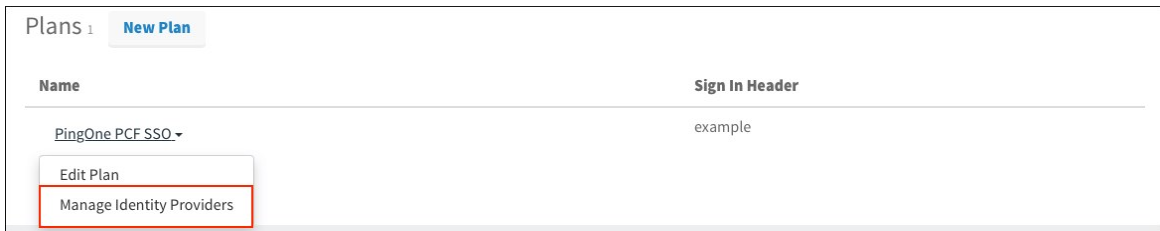


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

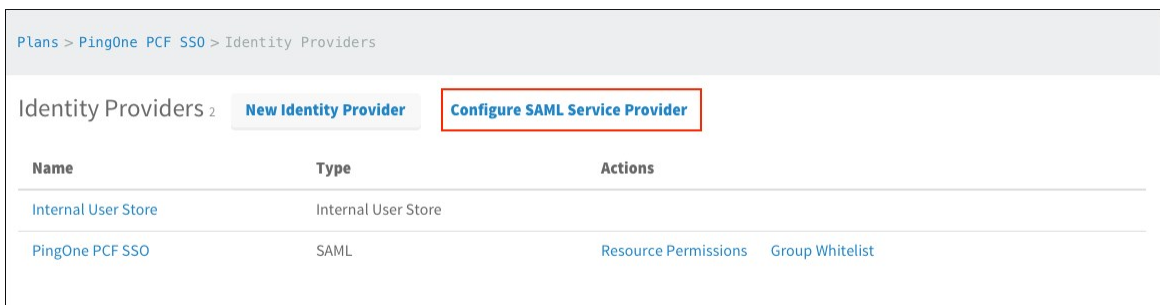
This topic describes how to set up PingOne Cloud as your identity provider by configuring SAML integration in both Pivotal Single Sign-On and PingOne Cloud.

Set up SAML in Single Sign-On

1. Log into the SSO Operator Dashboard at <https://p-identity.SYSTEM-DOMAIN> as a Plan Administrator.
2. Select your plan and click **Manage Identity Providers** on the dropdown.



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

[Download Metadata](#)

- ☒ Perform signed authentication requests
- ☐ Require signed assertions

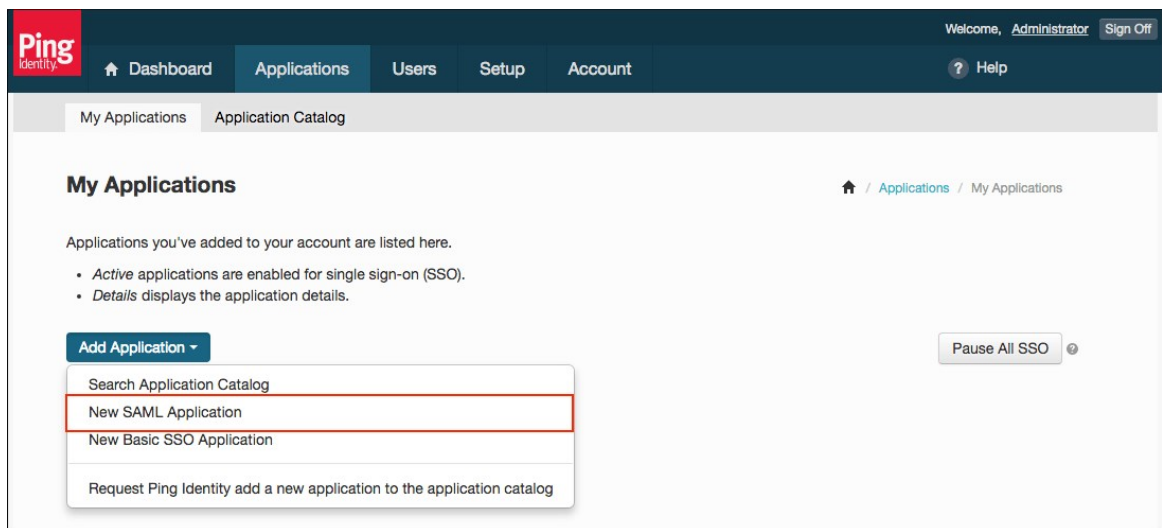
Cancel

Save

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

Set up SAML in PingOne Cloud

- Sign in as a PingOne Cloud administrator.
- Navigate to your app by clicking on **Apps**.
- Click **Add Application** and choose **New SAML Application**.



- Enter the **Application Name**, **Application Description**, **Category** and any **Graphics**.
- Click **Continue to Next Step** 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 ⓘ [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 ⓘ 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

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 **Continue to Next Step**.
7. (Optional) Under **SSO Attribute Mapping**, specify any app 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	<input type="text" value="firstName"/>	<input type="text" value="First Name"/> <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
2	<input type="text" value="lastName"/>	<input type="text" value="Last Name"/> <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
3	<input type="text" value="email"/>	<input type="text" value="Email"/> <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>
4	<input type="text" value="group"/>	<input type="text" value="memberOf"/> <input type="checkbox"/> As Literal <input type="button" value="Advanced"/>	<input type="checkbox"/> <input type="button" value="✕"/>

NEXT: Review Setup

8. Click **Save & Publish**.

9. Click **Finish**.

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

Configuring a Single Sign-On Service Provider

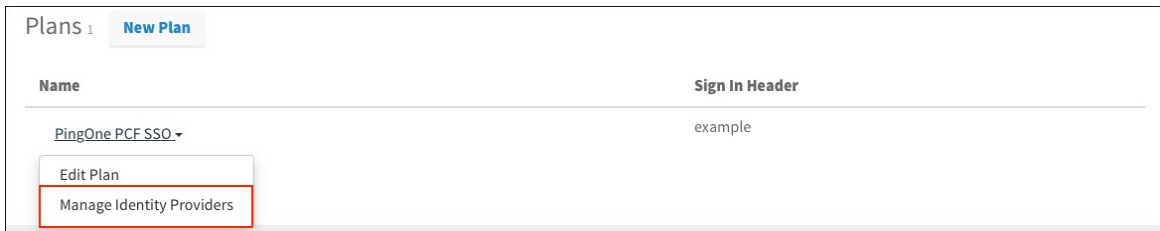


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

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

Set up SAML

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



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

New Identity Provider

Cancel **Create 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 2.0

Identity Provider Metadata

Identity Provider Metadata URL*

<https://idp.company.com/SAML2>

Fetch Metadata

▶ SAML File Metadata (optional)

Email Domains

Provide comma-separated list of domains for identity provider discovery

domain1.com, domain2.com

Advanced 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 **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 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: Pivotal Single Sign-On v1.11 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 Pivotal Single Sign-On and PingOne Cloud. An administrator can test both service provider and identity provider connections.

You can test your identity provider integration by deploying the [Pivotal Single Sign-On Service Sample Applications](#).

Test Your Service Provider Connection

1. Log in to Apps Manager at <https://apps.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 plan bound to your app. Click on the service instance 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...

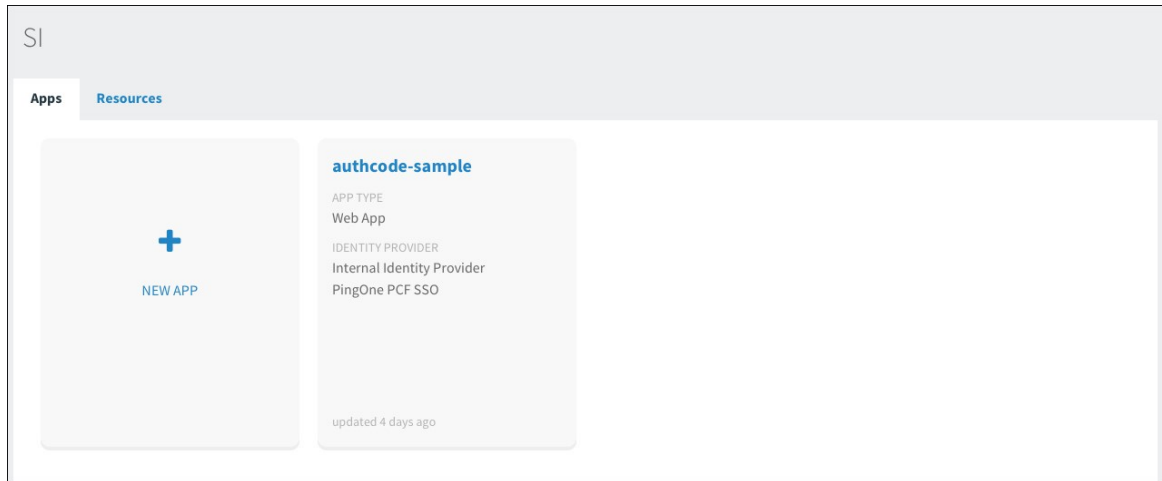
Services

Add Service

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



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



4. 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 app 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.

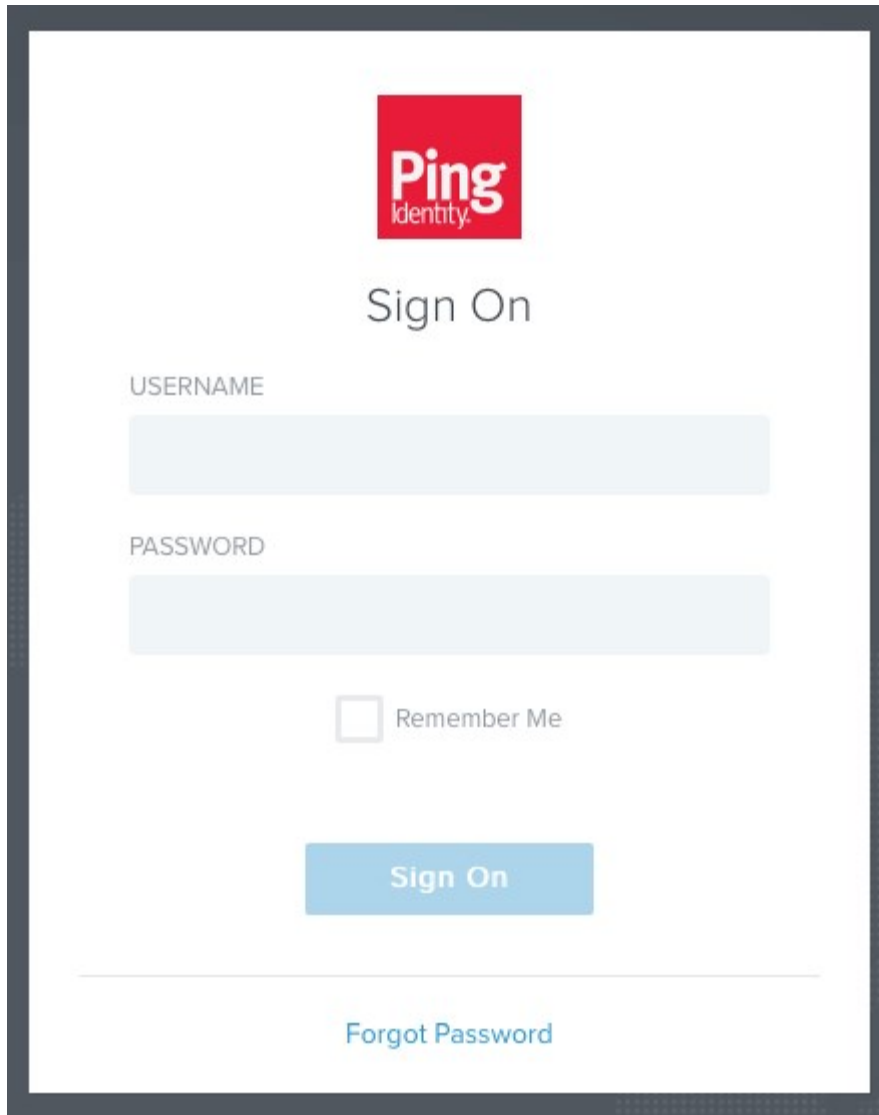
← → ↻ https://authcode-sample

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 On**.

The image shows a web page for Ping Identity Sign On. At the top center 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 gray font. Underneath this are two input fields: the first is labeled "USERNAME" and the second is labeled "PASSWORD". Both labels are in a small, dark gray font. Below the password field is a checkbox with the text "Remember Me" next to it. Further down is a blue button with the text "Sign On" in white. At the bottom of the form is a horizontal line, and below that is a link that says "Forgot Password" in a blue font.

8. The app 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" : "openid"
}
```

```

    "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?

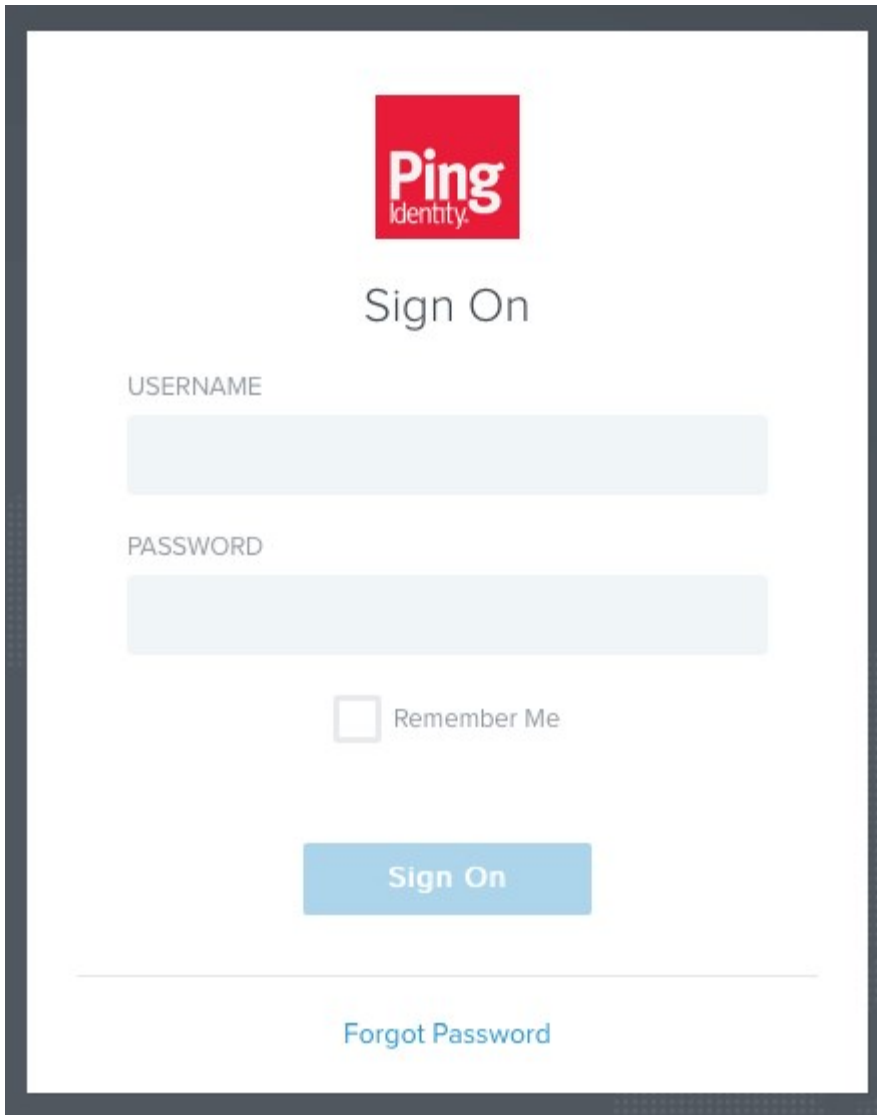
- [TODO List \(You need to configure the Resource Server sample app before using this\)](#)
- [See your account profile \(so you can de-authorize this client\)](#)
- [Log out](#)

Test Your Identity Provider Connection



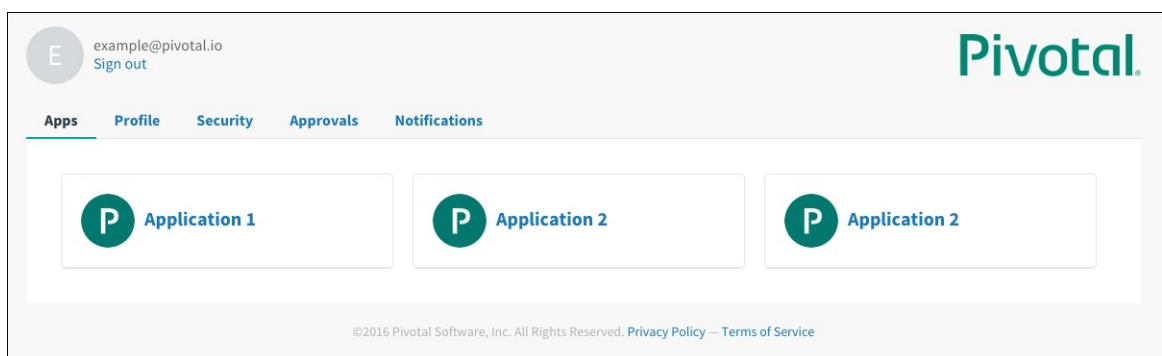
Note: Single Sign-On does not support identity provider-initiated flow into apps, but it does redirect the user to the User Account and Authentication (UAA) page to select apps assigned to the user.

1. Sign in to PingOne.



The image shows a web page for Ping Identity Sign On. At the top center 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 gray font. Underneath "Sign On" are two light blue input fields. The first field is labeled "USERNAME" in a small, dark gray font above it. The second field is labeled "PASSWORD" in a small, dark gray font above it. Below the password field is a checkbox with the text "Remember Me" to its right. Below the checkbox is a blue button with the text "Sign On" in white. At the bottom of the page, there is a horizontal line and the text "Forgot Password" in a blue font.

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



Test Your Single Sign-Off

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

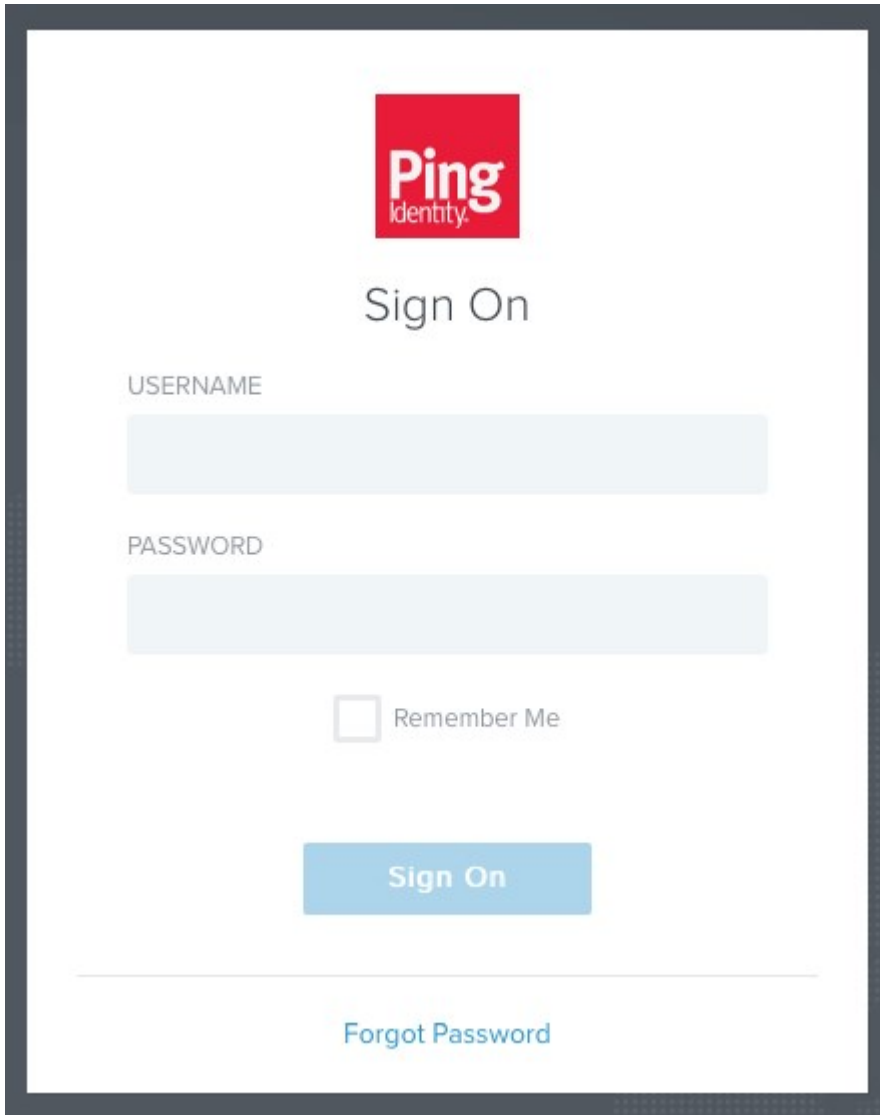
1. Sign into the sample app. Information about the access and ID token displays, as well as the **What do you want to do?** section.

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 PingOne login page.

The image shows the Ping Identity Sign On page. At the top center 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 grey font. Underneath "Sign On" are two input fields: the first is labeled "USERNAME" and the second is labeled "PASSWORD". Below the password field is a checkbox labeled "Remember Me". At the bottom of the form is a blue button with the text "Sign On" in white. Below the button is a horizontal line, and below that is a link that says "Forgot Password" in blue text.

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

Troubleshooting



Warning: Pivotal Single Sign-On v1.11 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.

Error

Symptom:

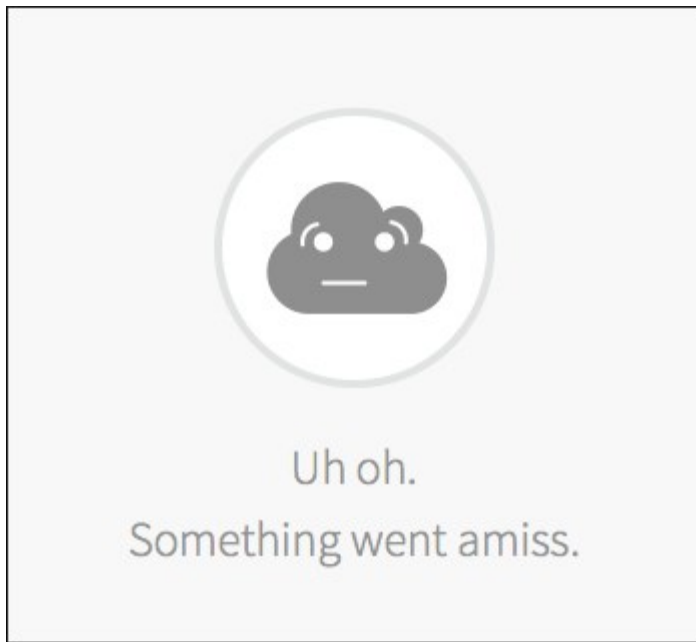


Explanations:

- 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 Single Sign-On 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 Single Sign- On 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 saml2-metadata-idp.xml

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



Warning: Pivotal Single Sign-On v1.11 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 Single Sign-On to integrate a Single Sign-On 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 Single Sign-On service plan to another using Single Sign-On.

Prerequisites

To integrate Plan-to-Plan OIDC with Single Sign-On, you must have the following:

- An active Single Sign-On service plan. This plan act as an identity provider.
- A second active Single Sign-On service plan. This plan 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 Pivotal Platform 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 Single Sign-On

Complete this process to set up Plan-to-Plan OIDC integration for the Single Sign-On 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 Single Sign-On, 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: Pivotal Single Sign-On v1.11 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 Pivotal 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).

Overview

A Plan-to-Plan OIDC integration enables users from the identity provider plan to authenticate into the relying party plan through OIDC.

To set up this integration:

1. [Meet the prerequisites](#)
2. [Set up relying party configurations in the identity provider plan](#)
3. [Set up the OIDC Identity Provider Configuration in the Relying Party Plan](#)
4. [Finish the configuration](#)

Prerequisites

You must meet the following prerequisites to set up Plan-to-Plan OIDC integration:

- 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.

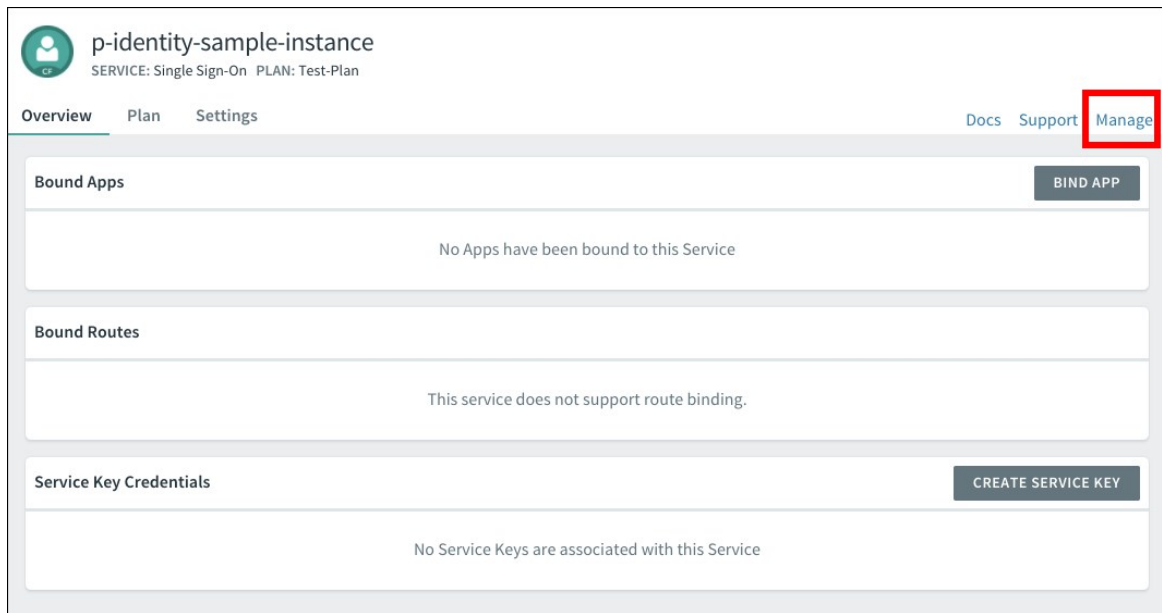
If you have not completed these prerequisites, see [Create or Edit Service Plans](#).

Set Up Relying Party Configurations in the Identity Provider Plan

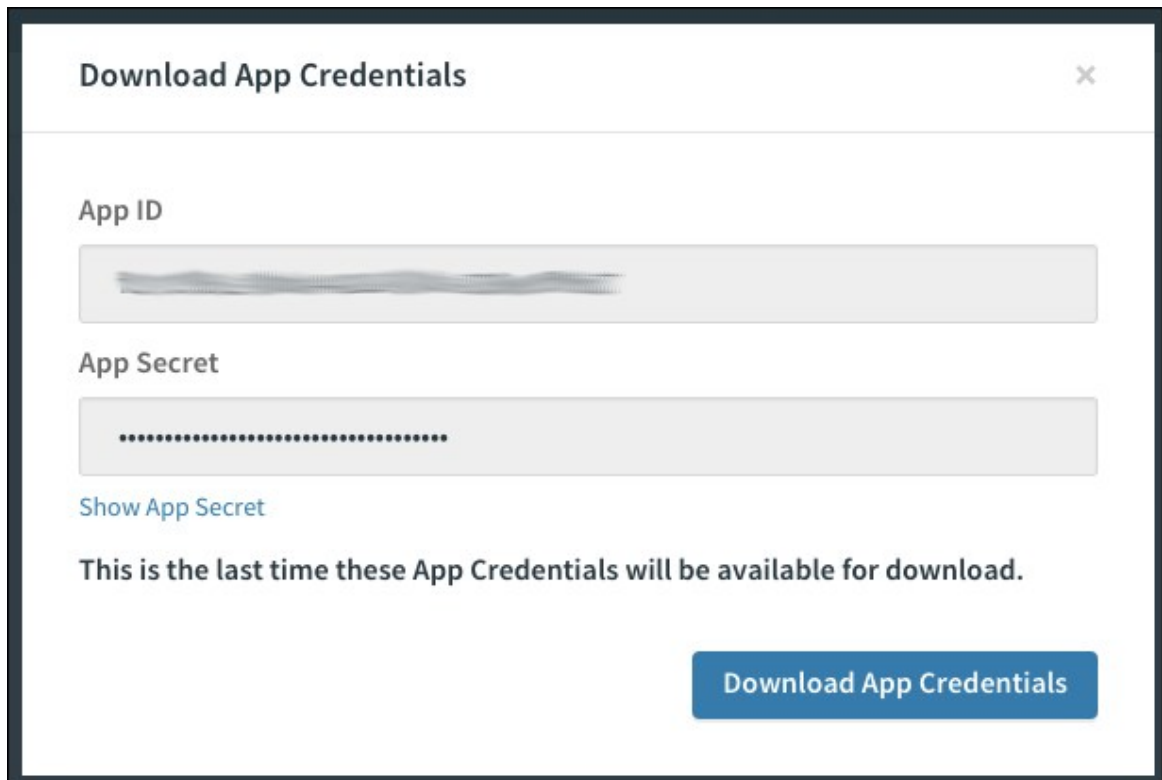
Follow the instructions below to set up relying party configurations in the identity provider plan.

1. Navigate to Apps Manager.
2. Select the space.
3. Click into the **Service** tab.

4. Click the service you want to modify.
5. Click **Manage**.



6. Click **New App**.
7. Type a name in the **App Name** field.
8. Choose **Web App** from the list of app types.
9. Type a temporary URL in the **Auth Redirect URIs** field. You replace this URL after configuring 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 prevent users from being prompted to authorize a login from the identity provider.
11. Click **Register App**. When the app is created successfully, you are 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

12. Click **Download App Credentials** to save the credentials for your app.



Warning: This is the last time you can download your app credentials. Pivotal recommends that you download the credentials and store them securely.

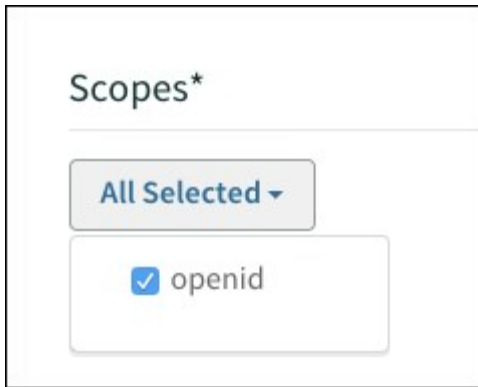
Set Up the OIDC Identity Provider Configuration in the Relying Party Plan

To set up the OIDC Identity Provider Configuration in the relying party plan, follow the steps below.

1. Follow steps 1 – 6 in [Add an OIDC Provider](#).
2. If you use a self-signed certificate for Pivotal Platform where the IDP is located, select the **Skip SSL Validation** checkbox. If you do not use a self-signed certificate, you can leave this box unchecked.
3. Select the **Enable Discovery** checkbox and type in the **Discovery Endpoint URL**.

This URL is `https://IDP-DOMAIN/.well-known/openid-configuration`, where `IDP-DOMAIN` is the domain setting you enter when you add the IDP service plan you are integrating.

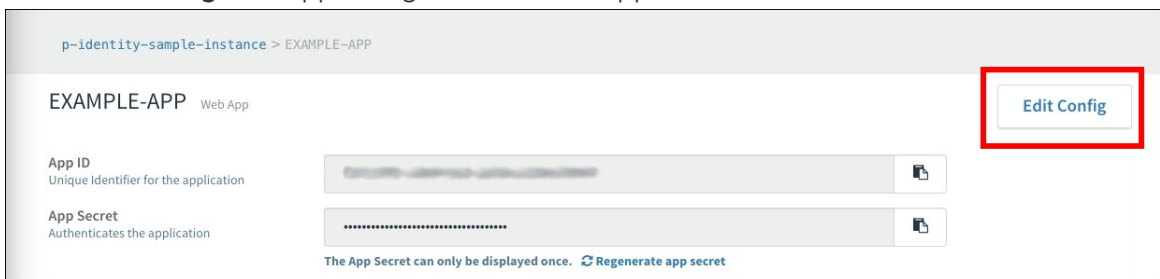
4. Fill in the **Relying Party OAuth Client ID** with the App Client ID from [the previous section](#).
5. Fill in the **Relying Party OAuth Client Secret** with the App Secret from [the previous section](#).
6. Confirm that `openid` is selected as a scope by clicking **All Selected**.



Finish Configuration

After you create an app, follow the steps below to finish configuration.

1. Return to the page for the app you created.
2. Click **Edit Config**. The app configuration screen appears.



3. Add an **Auth Redirect URL**. The URL should read `https://RELYING-PARTY-DOMAIN/login/callback/ORIGIN-KEY`

Where:

- ♦ `RELYING-PARTY-DOMAIN` is the domain setting you enter during Relying Party configuration.
- ♦ `ORIGIN-KEY` is based on the IDP name you set in the SSO Operator Dashboard.

4. Click **Save Config**.

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

Testing OIDC Integrations

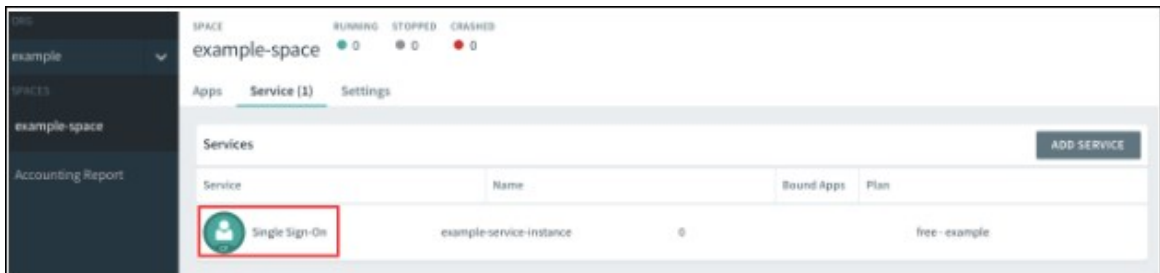


Warning: Pivotal Single Sign-On v1.11 is no longer supported because it has reached the End of General Support (EOGS) phase as defined by the [Support Lifecycle Policy](#). To stay up to date with the latest software and security updates, upgrade to a supported version.

This topic describes how a Pivotal Platform administrator can test the OpenID Connect (OIDC) connection between a Pivotal Single Sign-On service plan acting as an Identity Provider (IDP), and another Single Sign-On service plan acting as a Relying Party (RP).

Testing Your Single Sign-On Connection

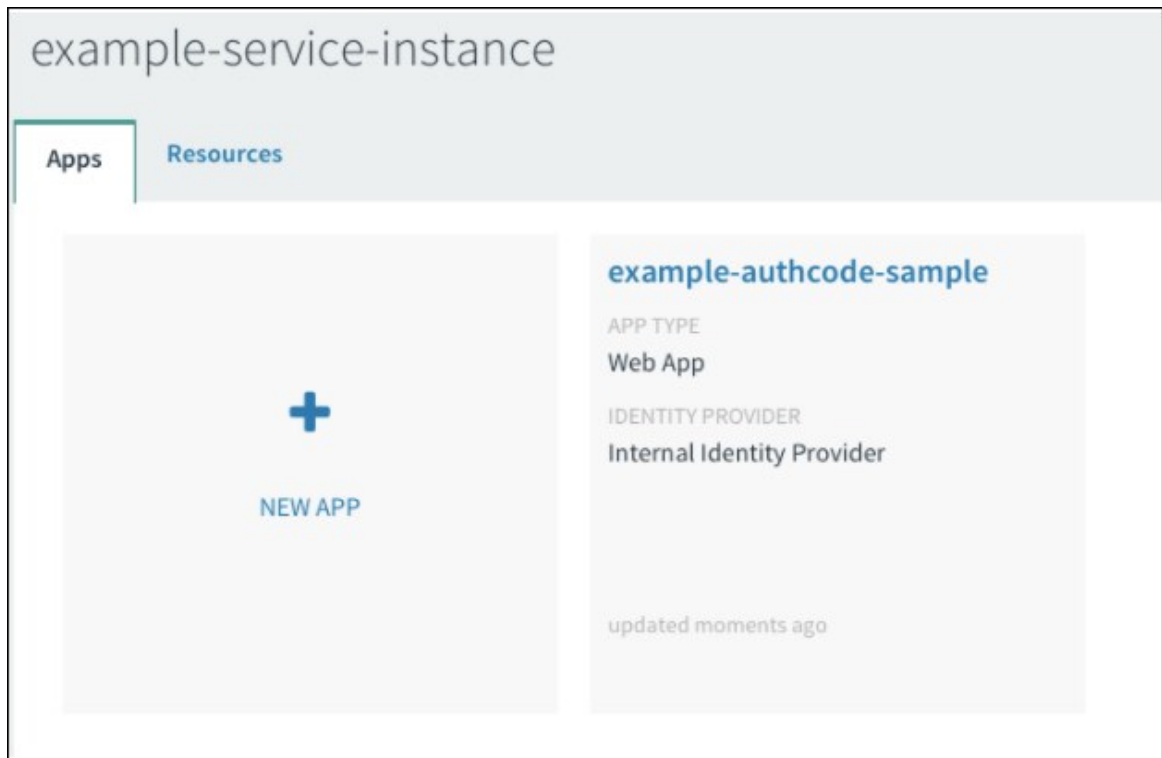
1. Log in to Apps Manager at <https://apps.SYSTEM-DOMAIN>.
2. Navigate to the org and space where your app is located.
3. Locate the service instance of the Single Sign-On plan bound to your app.



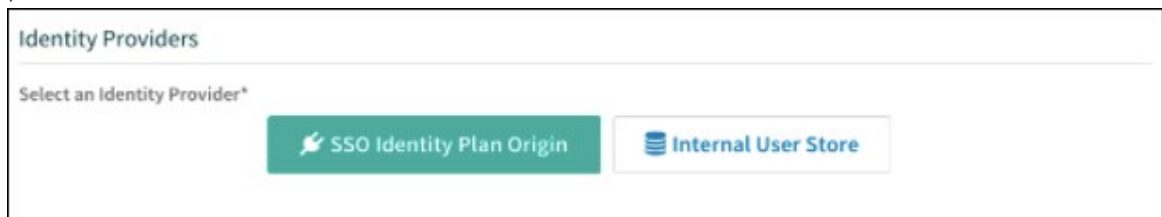
4. Select the service instance.
5. Click **Manage**.



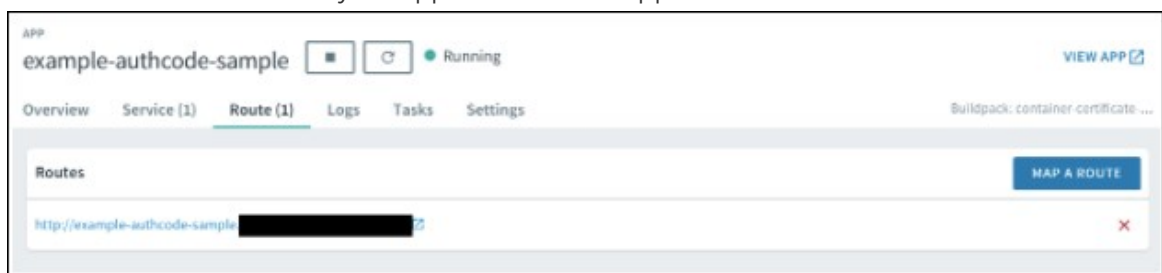
6. In the **Apps** tab, click your app.



7. Under **Identity Providers**, select the SSO Identity Plan Origin. Remove any other identity providers.



8. Return to Apps Manager.
9. Click the URL listed below your app to access the app.



10. Log in to the app. You will be redirected to the IDP to authenticate.
11. Sign in to the IDP.
12. If necessary, authorize the necessary scopes to connect the IDP with your app. If you need to do this, the IDP will prompt you.
13. 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](#)

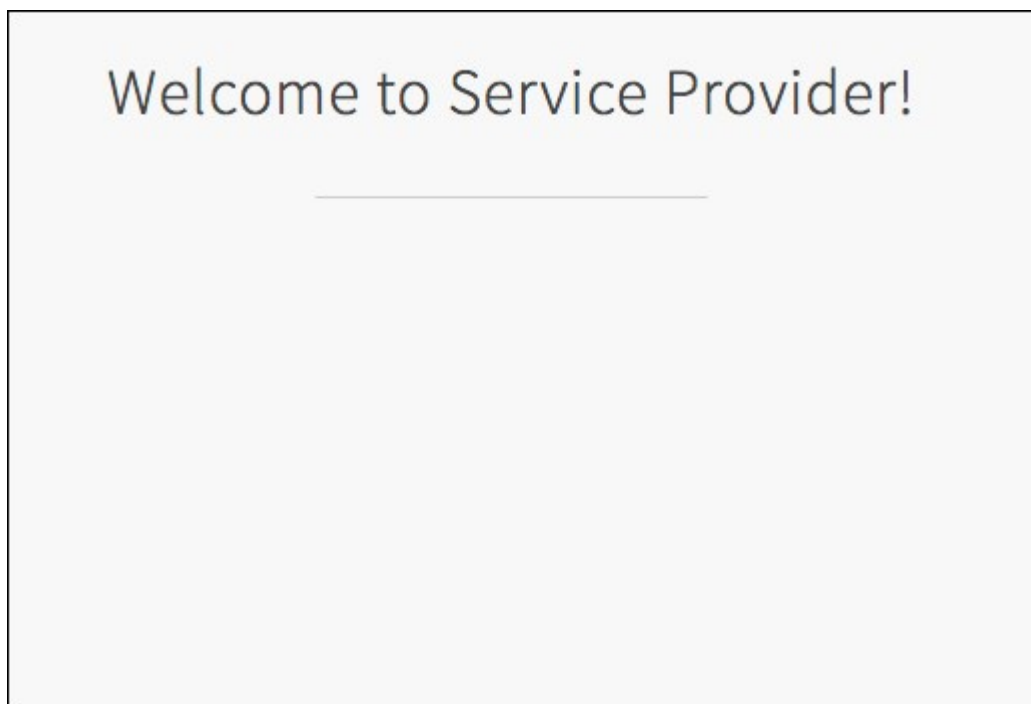
Troubleshooting Plan-to-Plan OIDC Integration



Warning: Pivotal Single Sign-On v1.11 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 partnership between two Pivotal Single Sign-On service plans, one acting as an Identity Provider (IDP) and one acting as a Relying Party (RP).

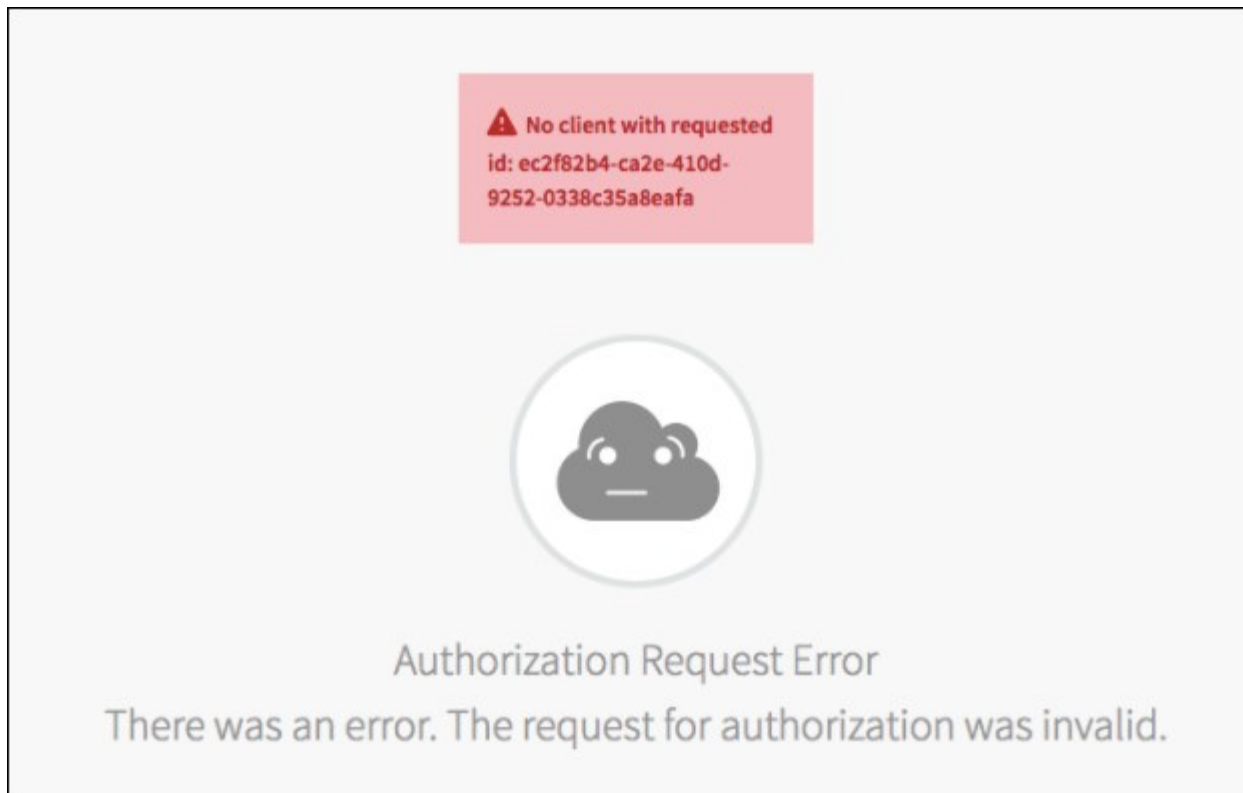
No link for OIDC, or the Service Provider Login page is blank



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 Pivotal Platform instance that uses a self-signed certificate.

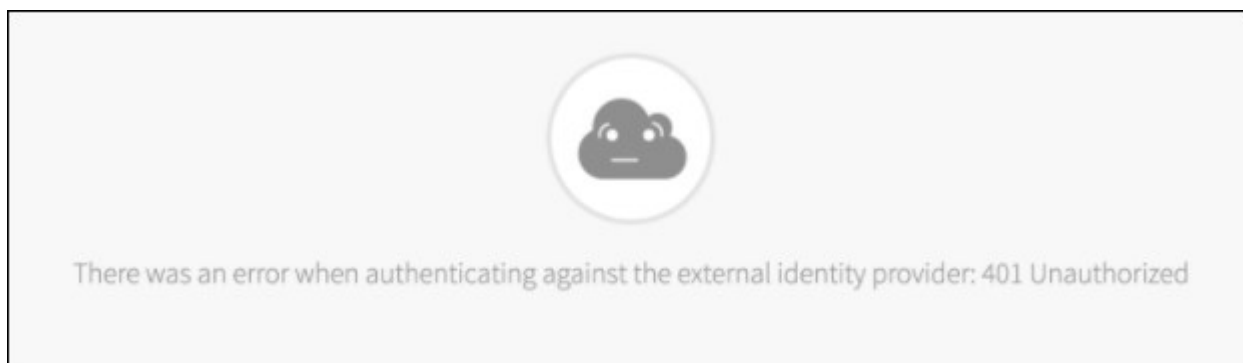
Authorization Request Error



Cause

You may have configured your OAuth client ID incorrectly.

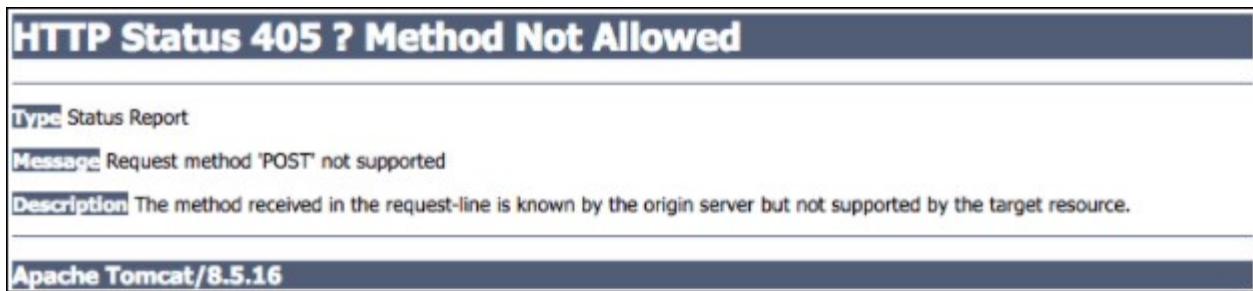
401 Unauthorized



Cause

You may have configured your OAuth client secret incorrectly.

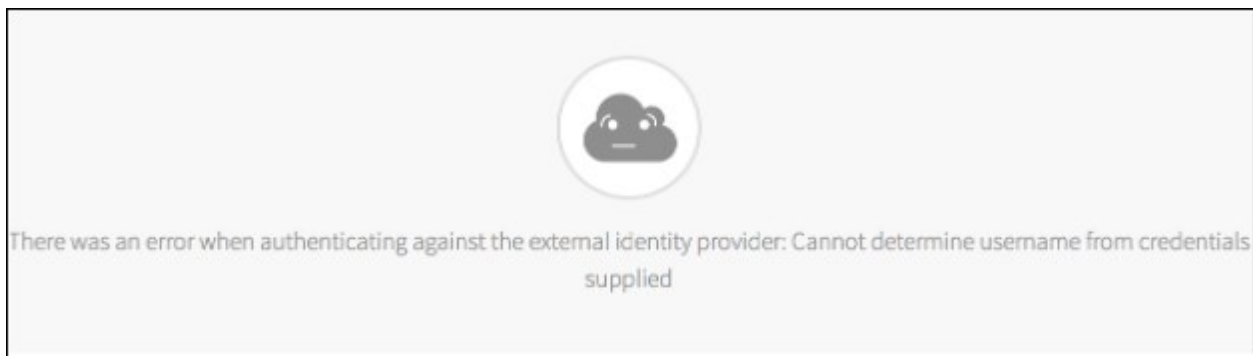
405 Method Not Allowed



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 Single Sign-On plan. Confirm that you are only requesting `openid` scopes.

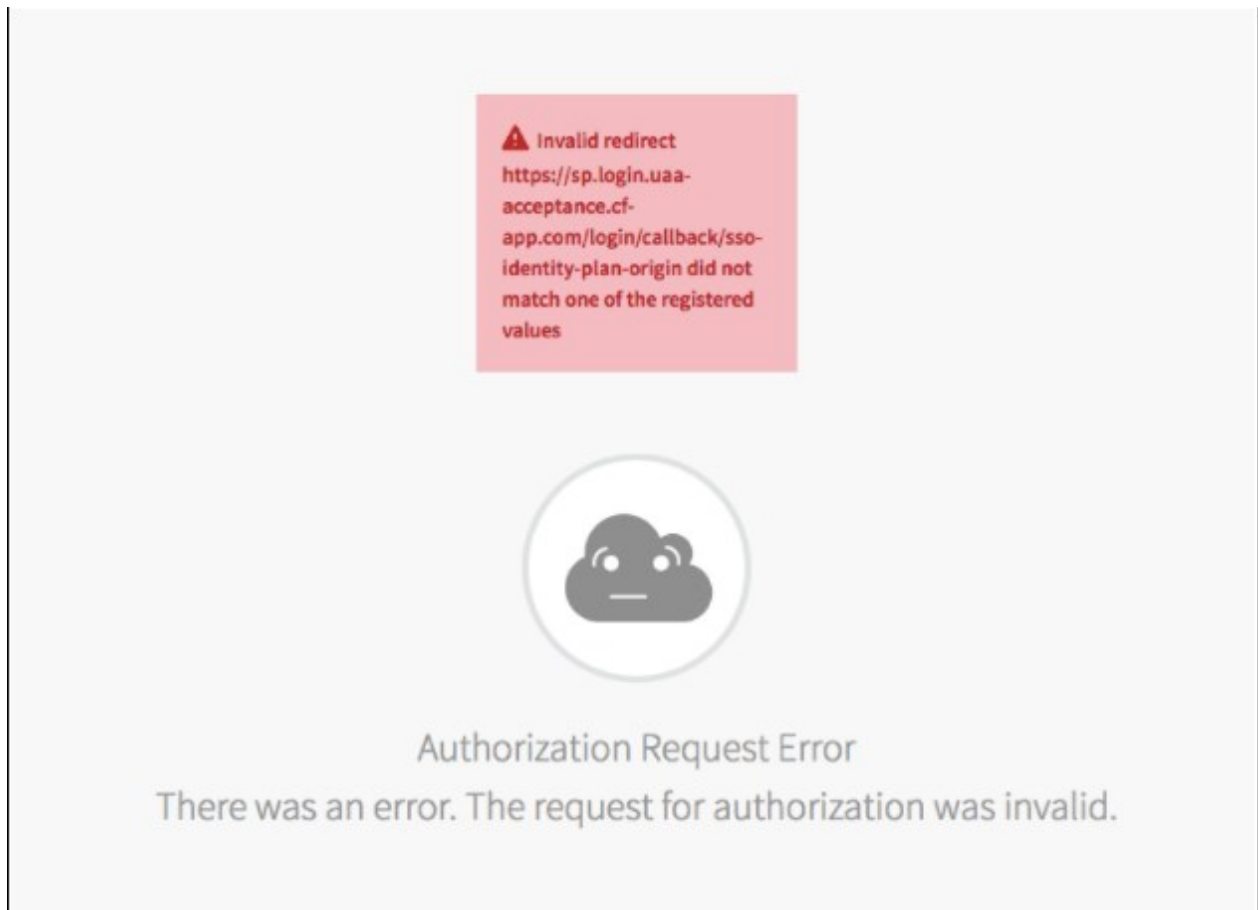
Cannot determine username with given credentials



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](#)