PingFederate Bridge

Contents

Opening the PingFederate Bridge administrative console....................22

Authentication..22

Applications...62

Security..100

Third parties

Amazon

AWS CloudHSM

incommon and oid

Java

JGroups and cluster

Log4j

Microsoft

Oracle

Red Hat

Splunk

Troubleshooting

Standards

RFCs

OASIS, WS, SAML

OAuth and OIDC

SCIM

PIC xref conref

quickly and easily configure user authentication from an on-premise directory to PingOne for Enterprise in the cloud.

PingFederate Bridge allows you to enable single sign-on (SSO) for PingOne for Enterprise and to integrate PingID multifactor authentication (MFA) with VPN using RADIUS. One or both of these options can be configured in a PingFederate Bridge environment.

To install and use PingFederate Bridge, you must have a PingOne for Enterprise account. If you do not have an account, you can register for one at *https://www.pingidentity.com/en/lp/d/p14e-trial.html* or use a registration code by going to *https://admin.pingone.com/web-portal/register*.

ⓘ **Important:**

PingOne for Enterprise must be set up with a PingFederate Bridge identity repository.

Installing PingFederate Bridge

Before you install PingFederate Bridge, make sure that your system meets the requirements and that you have the required ports available. You must also install Java.

System requirements

PingFederate Bridge supports the following system requirements. This section lists recommended versions and requirements.

Operating systems and virtualization

ⓘ **Note:**

PingFederate Bridge is tested with default configurations of operating-system components. If your organization customizes implementations or installs third-party plug-ins, deployment efforts might affect the PingFederate Bridge server.

Operating systems

- Amazon Linux 2
- Canonical Ubuntu 16.04 LTS
- Canonical Ubuntu 18.04 LTS
- Canonical Ubuntu 20.04 LTS
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Oracle Enterprise Linux 7.9 (Red Hat compatible kernel)
- Oracle Enterprise Linux 8.2 (Red Hat Compatible Kernel)
- Red Hat Enterprise Linux ES 7.9
- Red Hat Enterprise Linux ES 8.2
- SUSE Linux Enterprise 12 SP5
- SUSE Linux Enterprise 15 SP2

Docker support

- Docker version: 18.09.0 and later

  View PingFederate's Docker Image on *DockerHub*. Visit PingIdentity's DevOps *documentation* for more information. Note that only the PingFederate software is licensed under Ping Identity's end user license agreement, and any other software components contained within the image are licensed solely under the terms of the applicable open source/third party license.

  > ⓘ **Note:**
  >
  > Ping Identity accepts no responsibility for the performance of any specific virtualization software and in no way guarantees the performance or interoperability of any virtualization software with its products.

Virtualization

Although Ping Identity does not qualify or recommend any specific virtual-machine (VM) or container products other than those listed above, PingFederate Bridge has run well on several, including Hyper-V, VMWare, and Xen.

> ⓘ **Note:**
>
> The list of products is provided for example purposes only. We view all products in this category equally. Ping Identity accepts no responsibility for the performance of any specific virtualization software and in no way guarantees the performance, interoperability, or both of any VM or container software with its products.

Java environment

- Amazon Corretto 11
- Amazon Corretto 8
- OpenJDK 11
- Oracle Java SE Development Kit 11 LTS
- Oracle Java SE Runtime Environment (Server JRE) 8

> ⓘ **Note:**
>
> Ping Identity Java Support Policy applies. For more information, see Java Support Policy in the Ping Identity Knowledge Base.

Browsers

Runtime server

- Apple Safari
- Google Chrome
- Microsoft Edge
- Microsoft Internet Explorer 11
- Mozilla Firefox
- Apple iOS 13 (Safari)
- Google Android 10 (Chrome)

Administrative server

- Google Chrome

- Microsoft Internet Explorer 11
- Mozilla Firefox

> ⓘ **Note:**
>
> For a modern browser experience, we encourage our customers to migrate off of Microsoft Internet Explorer 11. For more information, including the end of support of Internet Explorer 11 in December, 2021, see *Upgrade considerations*.

TLS protocol

Runtime server and administrative server

- TLS 1.2 and 1.3

> ⓘ **Note:**
>
> TLS 1.3 requires Java 11.

Datastore integration

User-attribute lookup

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2
- Microsoft Active Directory 2016
- Oracle Unified Directory 12c
- Amazon Aurora (MySQL 5.6.10a)
- Amazon Aurora (PostgreSQL 10.11)
- Microsoft SQL Server 2016 SP2 and 2017
- Oracle Database 12c Release 1 (12.1.0.2.0)
- Oracle Database 19c
- Oracle MySQL 8.0
- PostgreSQL 9.6.19 and 11.9

SaaS or SCIM outbound provisioning

  Provisioning channel data source

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2
- Microsoft Active Directory 2016
- Oracle Unified Directory 12c

  Provisioning internal datastore

- Amazon Aurora (MySQL 5.6.10a)
- Amazon Aurora (PostgreSQL 10.11)
- Microsoft SQL Server 2016 and 2017
- Oracle Database 12c Release 1
- Oracle Database 19c
- Oracle MySQL 8.0
- PostgreSQL 9.6.19 and 11.9

SCIM inbound provisioning

- Microsoft Active Directory 2016
- Custom implementation through the PingFederate SDK

Just-in-time (JIT) inbound provisioning

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2
- Microsoft Active Directory 2016
- Oracle Unified Directory 12c
- Microsoft SQL Server 2016 SP2 and 2017

Account linking

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2
- Microsoft Active Directory 2016
- Oracle Unified Directory 12c
- Amazon Aurora (MySQL 5.6.10a)
- Amazon Aurora (PostgreSQL 10.11)
- Microsoft SQL Server 2016 SP2 and 2017
- Oracle Database 12c Release 1
- Oracle Database 19c
- Oracle MySQL 8.0
- PostgreSQL 9.6.19 and 11.9

OAuth client configuration and persistent grants

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2
- Microsoft Active Directory 2016
- Oracle Unified Directory 12c
- Amazon Aurora (MySQL 5.6.10a)
- Amazon Aurora (PostgreSQL 10.11)
- Microsoft SQL Server 2016 SP2 and 2017
- Oracle Database 12c Release 1
- Oracle Database 19c
- Oracle MySQL 8.0
- PostgreSQL 9.6.19 and 11.9
- Custom implementation through the PingFederate SDK

Registration and profile management of local identities

- PingDirectory 7.0, 7.2, 7.3, 8.0, 8.1, 8.2

Persistent authentication sessions

- PingDirectory 7.2, 7.3, 8.0, 8.1, 8.2
- Amazon Aurora (MySQL 5.6.10a)
- Amazon Aurora (PostgreSQL 10.11)
- Microsoft SQL Server 2016 SP2 and 2017
- Oracle Database 12c Release 1 and 19c
- Oracle MySQL 8.0
- PostgreSQL 9.6.19 and 11.9

ⓘ **Note:**

> PingFederate Bridge was tested with vendor-specific JDBC 4.2 drivers. For more information, see *Database driver information* on page 11.

Third-party cryptographic solutions

Hardware security modules (optional)

AWS CloudHSM

- Client software version: 3.2.1

> ⓘ **Note:**
>
> PingFederate Bridge must be deployed on one of the Linux operating systems supported by both AWS CloudHSM and PingFederate Bridge.

Thales Luna Network HSMs

- Universal Client 10.2

For more information about the Universal Client, including compatible HSMs, HSM firmware, appliance software, and client software, see the *documentation from Thales*.

Entrust nShield Connect (in FIPS 140-2 Level 3 mode)

- Host and Firmware version: 12.40.0
- Client driver version: 12.40.2
- Hardware Model: Net HSM 6000 appliance

> ⓘ **Note:**
>
> When integrating with a hardware security module (HSM), you must deploy PingFederate Bridge with Oracle Server JRE (Java SE Runtime Environment) 8 or Amazon Corretto 8.

Software cryptographic solution

Bouncy Castle

Bouncy Castle Java FIPS 1.0.2

Hardware requirements

Minimum hardware recommendations

- Multi-core Intel Xeon processor or higher

  4 CPU/Cores recommended
- 4 GB of RAM

  1.5 GB available to PingFederate Bridge
- 1 GB of available hard drive space

> ⓘ **Note:**

Although it is possible to run PingFederate Bridge on less powerful hardware, the guidelines provided accommodate disk space for default logging, auditing profiles, and CPU resources for a moderate level of concurrent request processing.

Database driver information

PingFederate Bridge is compatible with the following vendor-specific JDBC drivers.

PingFederate Bridge and JDBC driver compatibility

Database server	Driver information
Amazon Aurora (MySQL 5.6.10a)	**Driver version information** mysql-connector-java version 8.0.19 **Driver class** `com.mysql.cj.jdbc.Driver` **JDBC URL** `jdbc:mysql://`*`databaseservername`*`/`*`databasename`* **Database location** Regional **Database features** One writer and multiple readers
Amazon Aurora (PostgreSQL 10.11)	**Driver version information** postgresql version 42.2.5 **Driver class** `org.postgresql.Driver` **JDBC URL** `jdbc:postgresql://`*`databaseservername`*`/`*`databasename`* **Database features** One writer and multiple readers
Microsoft SQL Server 2016 SP2 and 2017	**Driver version information** sqljdbc version 7.2.1 **Driver class** `com.microsoft.sqlserver.jdbc.SQLServerDriver` **JDBC URL** `jdbc:sqlserver://`*`databaseservername`*`;databaseName=`*`databasename`*

Database server	Driver information

Oracle Database 12c Release 1 and 19c	**Driver version information**
	ojdbc7 version 12.1.0.2.0
	
	Driver class
	`oracle.jdbc.OracleDriver`
	
	JDBC URL
	`jdbc:oracle:thin:@databaseservername:databasename`

Oracle MySQL 8.0	**Driver version information**
	mysql-connector-java version 8.0.15
	
	Driver class
	`com.mysql.cj.jdbc.Driver`
	
	JDBC URL
	`jdbc:mysql://databaseservername/databasename`

PostgreSQL 9.6.1 and 11.2	**Driver version information**
	postgresql version 42.2.5
	
	Driver class
	`org.postgresql.Driver`
	
	JDBC URL
	`jdbc:postgresql://databaseservername/databasename`

For additional information about these drivers, contact the respective vendors.

Port requirements

The following table summarizes the ports and protocols that PingFederate Bridge uses to communicate with external components. This information provides guidance for firewall administrators to ensure the correct ports are available across network segments.

ⓘ **Note:**

Direction refers to the direction of the initial requests relative to PingFederate Bridge. Inbound refers to requests PingFederate Bridge receives from external components. Outbound refers to requests PingFederate Bridge sends to external components.

PingFederate Bridge required ports and protocols

Service	Protocol, direction, transport, default port	Source	Destination	Description
Administrative console	HTTPS, inbound, TCP, 9999	Browsers accessing the administrative console, REST calls to the administrative API, web service calls to the Connection Management Service. Applicable to the console node in a clustered PingFederate Bridge environment.	Administrative node	Used for incoming requests to the administrative console. Configurable in the `run.properties` file.
Administrative console	HTTPS, outbound, TCP, 443	Administrator accessing online documentation. Applicable to the console node in a clustered PingFederate Bridge environment.	docs.pingidentity.com	Used for accessing online documentation from the administrative console.

Service	Protocol, direction, transport, default port	Source	Destination	Description
Runtime engine	HTTPS, inbound, TCP, 9031 (and 9032 if configured)	Browsers accessing the runtime server for SSO or SLO; web service calls to the SSO Directory Service; REST calls to the OAuth Client Management Service, the OAuth Access Grant Management Service, the Persistent Grant Management API, and the Session Revocation API. Applicable to all runtime engine nodes in a clustered PingFederate Bridge environment.	Runtime engine nodes	Used for incoming requests to the runtime engine. Configurable in the `run.properties` file.
Cluster traffic	JGroups, inbound, TCP, 7600	PingFederate Bridge peer servers in a clustered PingFederate Bridge environment.	Administrative node and runtime engine nodes	Used for communications between engine nodes in a cluster when the transport mode for cluster traffic is set to TCP (the default behavior). Configurable in the `run.properties` file.
Cluster traffic	JGroups, inbound, TCP, 7700	PingFederate Bridge peer servers in a clustered PingFederate Bridge environment.	Administrative node and runtime engine nodes	Used by other nodes in the cluster as part of the cluster's failure-detection mechanism when the transport mode for cluster traffic is set to TCP (the default behavior). Configurable in the `run.properties` file.
Cluster traffic (if configured)	JGroups, outbound, TCP, 443	PingFederate Bridge peer servers in a clustered PingFederate Bridge environment.	Amazon Simple Storage Service (Amazon S3) or an OpenStack Swift server	Used by all nodes when the optional dynamic discovery mechanism is enabled.

Service	Protocol, direction, transport, default port	Source	Destination	Description
Cluster traffic	JGroups, inbound,UDP, 7601	PingFederate Bridge peer servers in a clustered PingFederate Bridge environment.	Administrative node and runtime engine nodes	Used for communications between engine nodes in a cluster when the transport mode for cluster traffic is set to UDP. By default, the transport mode is TCP. Configurable in the `run.properties` file.
PingOne for Enterprise integration (if configured)	HTTPS and secure WebSocket, TCP, 443	PingFederate Bridge Applicable to the console node in a clustered PingFederate Bridge environment.	pingone.com	Used for communications between PingFederate Bridge and PingOne for Enterprise for establishing and maintaining a managed SP connection to PingOne for Enterprise, monitoring of PingFederate Bridge from the PingOne admin portal, authenticating end users against the PingOne for Enterprise Directory.
Active Directory domains/ Kerberos realms (if configured)	Kerberos, outbound, TCP or UDP, 88	PingFederate Bridge	Windows domain controllers	Used for communications between PingFederate Bridge and Windows domain controllers for the purpose of Kerberos authentication.
reCAPTCHA (if configured)	HTTPS, outbound, TCP, 443	PingFederate Bridge	www.google.com/ recaptcha/api/ site verify	Used by the HTML Form Adapter when invisible reCAPTCHA from Google is enabled to prevent automated attacks.
Administration notification	SMTP, outbound, TCP, 25 (465 if SMTPS)	All nodes	SMTP server	Used to send notification messages for various events. For more information, see *Runtime notifications*.

> ⓘ **Note:**
>
> For PingID integration, see *PingID required domains, URLs, and ports*
>
> Depending on the integration kits deployed and the connecting third-party systems, such as email server or SMS service provider, additional ports may be required.

Installing Java

You must install a Java runtime on your server before installing PingFederate Bridge.

About this task
PingFederate Bridge has been tested with the following Java environments:

- Amazon Corretto 11
- Amazon Corretto 8
- OpenJDK 11
- Oracle Java SE Development Kit 11 LTS
- Oracle Java SE Runtime Environment (Server JRE) 8

ⓘ **Note:**

Ping Identity Java Support Policy applies. For more information, see Java Support Policy in the Ping Identity Knowledge Base.

ⓘ **Important:**

Due to the import restrictions of some countries, Oracle Server JRE (Java SE Runtime Environment) 8 has built-in restrictions on available cryptographic strength (key size). To use larger key sizes, the Java Cryptography Extension (JCE) "unlimited strength" jurisdiction policy must be enabled. For more information, see the *Java 8 release notes* from Oracle.

For Oracle Java SE Development Kit 11, the JCE jurisdiction policy defaults to unlimited strength. For more information, see the *Oracle JDK Migration Guide*.

Steps

1. Download and install a Java runtime.
2. Set the *JAVA_HOME* environment variable to the Java installation directory path and add its `bin` directory to the *PATH* environment variable.

Installing PingFederate Bridge on Windows

PingFederate Bridge is configured to run as a service; the service is started automatically at the end of the installation process.

Before you begin

Before you install PingFederate Bridge, make sure that you have the following in place:

- You must have a PingOne for Enterprise account and be connected to the PingFederate identity repository. For more information see *Connect to PingFederate Bridge*.
- You must be logged into a system with appropriate privileges to install and run an application.
- A supported version of the Java runtime environment must be installed as explained in *Installing Java* on page 16.

  The *JAVA_HOME* system environment variable must be set to the Java installation directory path. Also, the full path to the Java `bin` directory must be added to the Path system environment variable.

Steps

1. Download the PingFederate installer for Windows from PingOne for Enterprise.
2. Run the installer and follow the installation steps.

    When the installation has finished, you see the **Completed the PingFederate Setup Wizard** window.

3. Click the link to open the PingFederate administrative console in a browser where you can begin the initial setup.

ⓘ **Important:** On the PingFederate administrative console, select **Yes, Connect to PingOne for Enterprise** to begin your PingFederate Bridge setup.

Setting up PingFederate Bridge

The first time you open the PingFederate Bridge administrative console, you see the initial setup screen, from which you can begin setting up PingFederate Bridge. Each screen has online help, which you can access by clicking the question mark icon (?) located in the top right corner.

Connecting PingFederate Bridge to PingOne for Enterprise

Integrating PingFederate Bridge with PingOne for Enterprise provides a powerful solution combining the benefits of an on-premise deployment with the flexibility of a cloud solution.

Steps

1. Select **Yes, Connect to PingOne for Enterprise**, and then click **Sign on to PingOne to get your activation key**.
2. Sign on using your PingOne credentials.

    ⓘ **Note:**

    If you do not have a PingOne for Enterprise account, you can register for a free trial.

3. In the PingOne administrative portal, copy the value from the **Activation Key** field.
4. On the **PingOne Account** tab of the PingFederate administrative console, paste the value in the **Activation Key** field.
5. Click **Next**.

Connecting PingFederate Bridge to a directory server

On the **Identities** tab, choose whether to connect PingFederate Bridge to a directory server. If you connect to a directory server, LDAP is used for user lookup and credential validation.

About this task

You do not need to use a directory server if you are only using PingID VPN or if you are configuring PingFederate Bridge to function as a test identity provider. If this is the case, select **No, Don't Connect a Directory Server** and click **Next**.

To connect to a directory server:

Steps

1. Select **Yes, Connect a Directory Server**.

   Configuration fields appear.
2. Enter information in the fields that is appropriate for your directory server.

Field	Description
Directory Type	Select the type of directory server from the list.
	See *System requirements* on page 6 for a list of supported directory servers.
Data Store Name	Enter the name of the datastore.
Hostname	Enter the location of the directory server.
	It can be the IP address, the host name, or the fully qualified domain name of the directory server. The entry might include a port number.
Service Account DN	Enter the distinguished name (DN) of the service account that PingFederate can use to communicate with the directory server.
Password	Enter the password associated with the service account.
Search Base	Enter the DN of the location in the directory where PingFederate begins its datastore queries.
Search Filter	Enter the LDAP query to locate a user record for attribute lookup and potentially credential validation.
	The default value is either `sAMAccountName=${username}` or `uid=${username}`, depending on the selected directory type.
	If you require a more advanced search filter, ensure the value is a valid LDAP filter. For more information, consult your directory administrators.

3. Click **Next**.

   PingFederate Bridge tries to establish a secure (LDAPS) connection to the directory server.

Unsecure Connection or Certificate Error

You see the **Unsecure Connection** tab if your directory server does not support LDAPS. If you want to continue without a secure connection, click **Next**. Alternatively, you can click **Previous** to go back to the **Identities** tab and specify a different directory server.

You see the **Certificate Error** tab if the certificate presented by the directory server is not trusted by PingFederate Bridge. You can click **Choose Certificate** to import a certificate used by the directory server to establish a secure connection and then click **Next**. Alternatively, you can click **Previous** to go back to the **Identities** tab and specify a different directory server.

Use Cases

On the **Use Cases** tab, select what you plan to use PingFederate Bridge to do.

PingOne SSO

Select this option and click **Next** if you want to enable single sign-on with PingOne for Enterprise and your configured identity store. For more information about using PingOne SSO, see the *PingOne for Enterprise Administration Guide*.

> ⓘ **Note:**
>
> If you have not connected to an LDAP directory, the system configures test user accounts.

Optionally, you can enable Kerberos authentication (if you are using Active Directory). You can also enable user provisioning. To do this, select **Additional SSO Features**, and click **Begin**. After you have finished the Kerberos and provisioning configuration, you will be returned to the **Use Cases** tab.

PingID VPN (RADIUS)

Select this option and click **Next** if you want to configure PingFederate Bridge to integrate with a RADIUS server to support PingID multifactor authentication through your virtual private network. When you select this option, a **Begin** button appears. Click **Begin**.

For more information about using PingID VPN, see the *PingID Administration Guide*.

Kerberos Authentication

If you are using Active Directory, you see the **Kerberos Authentication** tab, on which you can optionally configure Kerberos-based authentication for Windows users.

About this task

If you do not want to configure Kerberos authentication, click **Next**.

To configure Kerberos authentication:

Steps

1. Select the **Configure Kerberos Authentication** check box to display the configuration fields.
2. Enter the following information:

   Realm Name

   > Enter the Kerberos realm name.

   Realm Username

   > Enter the Kerberos username.

Realm Password

Enter the Kerberos password.

Internal IP Ranges (At Least One Required)

Enter one or more internal IP ranges in CIDR notation to indicate the boundaries of your network, and click **Add**.

KDC Hostnames (Optional)

Optionally, enter a Kerberos Key Distribution Center (KDC) hostname and click **Add**. You can add multiple KDC hostnames. If you do not specify a hostname, PingFederate Bridge uses a DNS query to find a list of KDCs.

3. Click **Next**.

Basic Information

The **Basic Information** tab contains your base URL. Users will access this URL when interacting with PingFederate Bridge.

Steps

1. Change the base URL if needed.

The domain portion of the base URL should match the domain name of your organization because it is part of the address where your applications, users, and partners communicate with your PingFederate Bridge environment. You can update this URL as needed.

2. Click **Next**.

Confirmation

The **Confirmation** tab displays a summary of the configuration that will be applied to PingFederate Bridge.

Click **Next** to apply the configuration.

Complete

The **Complete!** tab congratulates you on successfully setting up PingFederate Bridge.

Steps

1. Make a note of the instructions under **What's Next?** You will need to complete these tasks when you begin configuring PingFederate Bridge.
2. Click **Done**.

Summary

The **Summary** tab contains a summary of your PingID VPN configuration.

Steps

1. Review your configuration.
2. If a setting is incorrect, click **Previous** and navigate to the information that needs to change. Make your changes and click **Next** until you see the **Summary** tab again.
3. When you are satisfied with your configuration, click **Done**.

Basic Settings

The **Basic Settings** tab contains configuration settings for connecting to your RADIUS server.

About this task

You see the **Validate User Credentials** and **PingID Username Attribute** fields only if you configured an LDAP directory on the **Identities** tab.

Steps

1. Enter information in the following fields:

Field	Description
Client IP	Enter the IP address of the VPN RADIUS client.
Client Shared Secret	Enter the password shared between PingFederate Bridge and the RADIUS server that is used to encrypt passwords.
Server Authentication Port	Enter the UDP port used to authenticate to the RADIUS server. Port number 1812 is provided by default.
Validate User Credentials	Enter the **Validate with LDAP** option if you want to use LDAP to validate credentials. This option is displayed only when an LDAP directory is connected.
PingID Username Attribute	Enter the LDAP attribute that represents the user identifier in PingID.

2. Click **Next**.

Provisioning

On the **Provisioning** tab, you can configure the provisioning of users from your directory server to PingID.

About this task

In this configuration, you specify the group where PingFederate Bridge should look for member users and update PingID when their email address, first name, or last name has changed. When PingFederate Bridge detects that a user has been removed from the specified group or disabled in the directory server, it sends an update to PingID to disable the PingID service for that account.

Steps

1. Select **Configure Provisioning**.
2. Under **PingID Group DN**, enter the distinguished name (DN) of the group for which you are configuring user provisioning.

The specified group must reside under the hierarchy of the previously defined **Search Base** value (as described in).
3. Select **Nested** if you want PingFederate Bridge to monitor changes for users through nested group membership.
4. Click **Next**.

Summary

The **Summary** tab contains a summary of your single sign-on configuration.

About this task

Steps

1. Review your configuration.
2. If a setting is incorrect, click **Previous** and navigate to the information that needs to change. Make your changes and click **Next** until you see the **Summary** tab again.
3. When you are satisfied with your configuration, click **Done**.

Provisioning

On the **Provisioning** tab, you can configure the provisioning of users from your directory server to PingOne for Enterprise.

About this task

In this configuration, you specify the group where PingFederate Bridge should look for member users and update PingOne for Enterprise when their email address, first name, or last name has changed. When PingFederate Bridge detects that a user has been removed from the specified group or disabled in the directory server, PingFederate Bridge sends an update to PingOne for Enterprise to disable the PingOne for Enterprise service for that account.

Steps

1. Select **Configure Provisioning**.
2. Under **Group DN**, enter the distinguished name (DN) of the group for which you are configuring user provisioning.

   The specified group must reside under the hierarchy of the previously defined **Search Base** value (as described in *Connecting PingFederate Bridge to a directory server* on page 18).
3. Select **Nested** if you want PingFederate Bridge to monitor changes for users through nested group membership.
4. Click **Next**.

Opening the PingFederate Bridge administrative console

The PingFederate Bridge administrative console is built around a system of wizard-like control screens, in which you configure various settings and components to support your federation use cases.

Steps

1. Make sure that the PingFederate Bridge service is running.
2. Start a web browser.
3. Browse to the following URL:

   ```
   https://pf_host:9999/pingfederate/app
   ```

   where pf_host is the network address of your PingFederate Bridge server. It can be an IP address, a host name, or a fully qualified domain name. It must be reachable from your computer.

Authentication

On the **Authentication** tab, you can configure several features related to integration and policies.

These features include:

- IdP adapters

- Authentication API applications
- IdP Default URL
- Policies
- Selectors
- Policy contracts
- Sessions

Integration

Under **Authentication**# **Integration**, you can configure IdP adapters, authentication API applications, and the IdP Default URL.

See the following sections:

- *Managing IdP adapters* on page 23
- *Managing authentication applications* on page 37
- *Configuring a default URL and error message* on page 39

Managing IdP adapters

An identity provider (IdP) adapter looks up session information and provides user identification to PingFederate Bridge. You must configure at least one instance of an IdP adapter in order to set up connections to service provider (SP) partners.

About this task

PingFederate Bridge comes bundled with the PingID integration kit and the following adapters:

- Composite Adapter
- HTML Form IdP Adapter
- HTTP Basic IdP Adapter
- Identifier First Adapter
- Kerberos Adapter
- OpenToken IdP Adapter
- PingID Adapter
- Reference ID IdP Adapter
- X.509 Certificate IdP Adapter

Additional integration kits are available on the PingFederate *download page* under the **Add-ons** tab.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**.
2. In the **Manage IP Adapter Instances** window, choose from the following options.

Option	Description
Create New Instance	Configure a new instance
<Existing instance link> **under Instance Name**	Modify an existing instance
Check Usage	Review the usage of an existing instance

Option	Description
Delete or Undelete	Remove an existing instance or cancel the removal

> ⓘ **Note:**
>
> By default, PingFederate Bridge automatically checks multi-connection errors whenever you access this window. This verifies that configured connections are not adversely affected by changes made here.
>
> If you experience noticeable delays in accessing this page, you can disable automatic connection validation. Go to **System**# **Server**# **General Settings**.

Creating an IdP adapter instance
The first step in creating an adapter instance is choosing an adapter type.

Steps

- On the **Type** tab, configure the basics of this adapter instance.
  a. Enter the instance name and ID.
  b. From the **Type** list, select the adapter type.
  c. Optional: From the **Parent Instance** list, select an existing type.

    If you are creating an instance that is similar to an existing instance, you might consider making it a child instance by specifying a parent. A child instance inherits the configuration of its parent unless overridden. You can specify overrides during the rest of the setup.

Configuring an IdP adapter instance
You can configure existing instances of IdP adapters.

About this task

Depending on the selected adapter, the **IdP Adapters** window presents you with different configuration parameters.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**.
2. Click the IdP adapter instance you want to configure.
3. Follow the in-product instructions to configure the adapter instance.

> ⓘ **Note:**
>
> If this is a child instance, select the override check box to modify the configuration.

If you are configuring one of the bundled adapters, see *Bundled adapters* or *Integrate with PingID for PingFederate SSO* for configuration information.

If you are configuring an adapter from an integration kit, including any software as a service (SaaS) connector, go to *Integrations Overview*, locate the adapter's documentation, and configure the adapter instance accordingly.

Extending an IdP adapter contract
You can use the **Extended Contract** tab to extend the contract of existing IdP adapters, especially those using the Composite Adapter.

About this task

If you are using the Composite Adapter, you must add attributes from the IdP adapter instances that comprise the composite configuration in this tab. For more information, see *Composite Adapter*.

If the adapter does not return values for the extended attributes, or if you prefer to fulfill them differently using datastore queries, dynamic text values, or results from OGNL expressions, you can define their fulfillment on the **Adapter Contract Mapping** tab. For more information, see *Defining the IdP adapter contract* on page 26.

ⓘ **Note:**

If this is a child instance, select the override check box to modify the configuration.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**.
2. Click the existing IdP adapter instance you want to modify, and then click the **Extended Contract** tab.
3. Enter the name of the desired attribute. Click **Add**.
4. Repeat as needed to add more attributes.
5. To save changes, click **Done**.

Setting pseudonym and masking options
You can set pseudonym and masking options to uniquely identify a user to your PingFederate Bridge SP partners.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters** and click the IdP adapter instance that you want to change.
2. On the **Adapter Attributes** tab, do the following:
   a. Optional: From the **Unique User Key Attribute** list, select an attribute to uniquely identify users signing on with this adapter. The attribute's value will be used to identify user sessions across all adapters. **None** is selected by default.

   ⓘ **Note:**

   If you choose a custom user key attribute, PingFederate Bridge uses the value of the attribute after the Adapter Contract Mapping (if any) has been evaluated. If you choose a custom user key

attribute that is based on the username, configure the adapter's password credential validators to trim spaces.

ⓘ **Important**:

For the HTML Form Adapter, If you enabled the **Revoke Sessions after Password Change or Reset** option in the **IdP Adapter** tab, you cannot select **None** as the unique user key attribute. Doing so will result in an error message.

---

b. Select the check box under **Pseudonym** for the user identifier of the adapter and optionally for the other attributes, if available.

    This selection is used if any of your service provider (SP) partners use pseudonyms for account linking.

ⓘ **Note:**

A selection is required whether or not you use pseudonyms for account linking. This allows account linking to be used later without having to delete and reconfigure the adapter. Ensure that you choose at least one attribute that is unique for each user, such as a user's email, to prevent assigning the same pseudonym to multiple users.

---

c. Select the check box under **Mask Log Values** for any attributes whose values you want PingFederate Bridge to mask in its logs at runtime.

ⓘ **Note:**

Masking is not applied to the unique user key attribute in the logs even though the attribute used for the key is marked as **Mask Log Values**.

---

d. Select the **Mask all OGNL-expression generated log values** check box, if OGNL expressions might be used to map derived values into outgoing assertions and you want those values masked.

Defining the IdP adapter contract
You can change the default identity provider (IdP) adapter contract settings using the **Adapter Contract Mapping** tab.

About this task

An IdP adapter contract is a contract that can be used to fulfill the attribute contract passed to your service provider (SP) partners. By default, PingFederate Bridge fulfills the IdP adapter contract with attribute values from the adapter. You can optionally configure PingFederate Bridge to fulfill the IdP adapter contract with attribute values from local datastores, dynamic text values, results from OGNL expressions, or a combination of them. In addition, you can verify requests using the Token Authorization framework.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**
2. Click the **Instance Name** of the existing IdP adapter instance you want to configure.
3. Go to the **Adapter Contract Mapping** tab.

ⓘ **Note:**

If this is a child instance, select the **Override Adapter Contract** check box to modify the configuration unless you have already selected the override option in the **Extended Contract** tab, in which case the **Override Adapter Contract** check box is automatically selected for you.

---

4. Click **Configure Adapter Contract**.

   ▪ For information on **Attribute Sources & User Lookup**, see *Defining attribute sources and user lookup* on page 27.
   ▪ For information on **Adapter Contract Fulfillment**, see *Configuring IdP adapter contract fulfillment* on page 33.
   ▪ For information on **Issuance Criteria**, see *Defining issuance criteria for IdP adapter contract* on page 34.

5. On the **Summary** tab, click **Done** to save your adapter contract configurations, or **Cancel** to discard them.

Defining attribute sources and user lookup

From the **Attribute Sources & User Lookup** window, you can add new attribute sources or manage existing attribute sources to supply attributes for the identity provider (IdP) adapter contract or the token authorization framework.

About this task

Attribute sources are specific datastore or directory locations containing information that might be needed for the IdP adapter contract or the token authorization framework. You can use more than one attribute source when mapping values to the IdP adapter contract.

For more information about token authorization, see *Token authorization*.

The PingFederate Bridge IdP server supports separate datastores to look up attributes for a single mapping. For example, you can query multiple datastores for values and map those values in one mapping, or query a datastore for a value and use that value as input for subsequent queries of other datastores.

Queries are executed in the order they are displayed on the **Attribute Sources & User Lookup** tab. Use the up and down arrows as needed to adjust the order.

If a required attribute, such as the user identifier username for the HTML Form Adapter or subject for the OpenToken IdP Adapter, cannot be fulfilled, the request fails.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**
2. Click the name of the existing IdP adapter instance you want to configure in the **Instance Name** list.
3. Click the **Adapter Contract Mapping** tab.

   > ⓘ **Note:**
   >
   > If this is a child instance, select the **Override Adapter Contract** check box to modify the configuration unless you have already selected the override option in the **Extended Contract** tab, in which case the **Override Adapter Contract** check box is automatically selected for you.

4. Click **Configure Adapter Contract**.
5.
   ▪ If your use case requires only dynamic texts or results from OGNL expressions without any attributes from local datastores, skip to *Configuring IdP adapter contract fulfillment* on page 33.
   ▪ To add an attribute source, click **Add Attribute Source**.
   ▪ To modify an existing instance, select it by name under **Description**.
   ▪ To remove an existing instance or to cancel the removal request, click **Delete** or **Undelete** under **Action**.

Choosing a datastore
This topic describes how to select a datastore for PingFederate Bridge to use for looking up attributes.

About this task

On the **Data Store** window, choose a datastore for PingFederate Bridge to look up attributes.

Steps

1. Enter a description and ID, if prompted, for the datastore.
2. Select a datastore instance from the **Active Data Store** list.

> ⓘ **Tip:**
>
> If the datastore you want is not shown in the **Active Data Store** list, click **Manage Data Stores** to review or add a datastore instance.

3. Depending on the datastore type, the rest of the setup varies as follows.

Data store type	Required tasks
JDBC	▪ *Specifying database tables and columns* on page 31 ▪ *Entering a database search filter*
LDAP	▪ *Specifying directory properties and attributes* on page 29 ▪ *Defining encoding for binary attributes* (optional) ▪ *Entering a directory search filter* on page 30
Other	▪ *Specifying data source filter and fields*

Entering an LDAP directory search filter
On the **LDAP Filter** tab, enter a filter for PingFederate Bridge to query the data you selected to retrieve a record associated with a particular value from the user's session.

About this task

The filter is in the form:

```
attribute1=value1
```

The left side (*attribute1*) is an attribute from your directory.

> ⓘ **Tip:**
>
> To see a list of attributes, click the **View List of Available LDAP Attributes** link.

The right side (*value1*) is the match-against value, generally a variable passed in from either an authentication source for an identity provider (IdP) or an assertion for a service provider (SP). The variables are shown underneath the **Filter** text field. If you are retrieving attributes from multiple data stores using one mapping, attributes available from other sources, if previously configured, are listed near the bottom of the window.

You can also apply additional search criteria by using other attributes from the target object class.

A filter narrows a search to locate requested data by either including or excluding specific records. A filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using three components: at least one attribute (attribute data type) on

which to search, a search filter operator that will determine what to match, and the value of the attribute being sought.

Steps

1. On the **LDAP Filter** window, enter a search filter in the text field.
2. Ensure the syntax and variable names are correct. For general information about search filters, consult your directory documentation.
3. Click **Next** to complete the configuration to query attributes from the directory server.

   Later in the workflow, you can use the attribute values returned from your directory server in the applicable contract fulfillment window, the issuance criteria window, or both, to fulfill your use case.

Example

Suppose you want to locate user records by matching the mail Active Directory (AD) user attribute against an extended attribute, eml, in your access token contract for the purpose of mapping attributes to an OpenID Connect policy. As a passed-in variable from the access token, `${eml}` is shown underneath the **Filter** text field.

On the **LDAP Filter** window, enter the following filter in the **Filter** text field.

```
mail=${eml}
```

mail

   An AD user attribute containing the email address of the user

${eml}

   The value of the extended attribute (eml) in the access token contract

---

ⓘ **Important:**

You must use the `${}` syntax to retrieve the value of the enclosed variable.

Specifying directory properties and attributes
Use these instructions to initiate ways to specify methods for PingFederate Bridge to search for particular user data.

About this task

On the **LDAP Directory Search** window, specify the branch of your directory hierarchy where you want PingFederate Bridge to look up user data. For more information about each field, refer to the following table.

Field	Description
Base DN	The base distinguished name (DN) of the tree structure in which the search begins. This field is optional if records are located at the root of the directory.
Search Scope	The node depth of the query. Select **Subtree** (the default value), **One level** or **Object**.
Root Object Class	The object class containing the desired attributes.
Attributes	A list of attributes based on the selected **Root Object Class** value.

Steps

1. Optional: Specify a base DN.

2. Select a search scope.
3. Optional: Click **View Attribute Contract** to determine what attributes to look up.
4. Select a root object class and an attribute, and then click **Add Attribute**.

> ⓘ **Note:**
>
> You do not have to add an attribute here to use it as part of a search filter. Add only the attributes that are required by subsequent sibling configuration items, such as contract fulfillment or token authorization. Any added attributes that are left unused are removed when the configuration is saved.

-   Microsoft Active Directory

    If you choose the memberOf attribute, an optional check box, **Nested Groups**, appears on the right. Select this check box if you want PingFederate Bridge to query for groups the end users belong to directly and indirectly through nested group membership (if any) under the base DN.

    For example, if you have three groups under a base DN: Canada, Washington and Seattle. Seattle is a member of Washington. Ana Smith is an end user and a member of Seattle. If the **Nested Groups** check box is selected, when PingFederate Bridge queries for Ana's memberOf attribute values, the expected results are Seattle and Washington. When the **Nested Groups** check box is not selected (the default), the expected result is Seattle.

-   Oracle Directory Server or Oracle Unified Directory

    Choose isMemberOf under **Attribute** for nested group membership. For information related to Oracle Directory Server, go to *docs.oracle.com/cd/E29127_01/doc.111170/e28967/ ismemberof-5dsat.htm*. For information related to Oracle Unified Directory, go to *Fusion Middleware Administering Oracle Unified Directory* and search for *memberof user attributes*.

> ⓘ **Tip:**
>
> If you need to include tokenGroups as one of the attributes, select **Object** as the search scope and enter a base DN matching the subject DN of the authenticated user—you can use variables from the authentication source (an adapter or an authentication policy contract) or results from the previous lookup in the base DN to fulfill this requirement.

5. Repeat step 4 to add more attributes as needed.

Example

Suppose you want to map the sn Active Directory (AD) user attribute into an OpenID Connect policy. The users for this use case reside under a specific container on your directory server, `OU=West, DC=example, DC=com`.

On the **LDAP Directory Search** window, enter `OU=West, DC=example, DC=com` as the base DN, keep the default **Search Scope** value (**Subtree**), select **<Show All Attributes>** from the **Root Object Class** list, select the `sn` AD user attribute, and click **Add Attribute**.

Entering a directory search filter
You can use a filter in PingFederate Bridge to query your selected data and retrieve a record associated with it.

About this task

On the **LDAP Filter** window, enter a filter for PingFederate Bridge to query the data you selected. The filter is in the form:

```
attribute1=value1
```

The left side (*attribute1*) is an attribute from your directory.

> ⓘ **Tip:**
>
> To see a list of attributes, click the **View List of Available LDAP Attributes** link.

The right side (*value1*) is the match-against value, generally a variable passed in from either an authentication source for an identity provider (IdP) or an assertion for a service provider (SP). The variables are shown underneath the **Filter** text field. If you are retrieving attributes from multiple data stores using one mapping, attributes available from other sources, if previously configured, are listed near the bottom of the window.

You can also apply additional search criteria by using other attributes from the target object class.

A filter narrows a search to locate requested data by either including or excluding specific records. A filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using three components: at least one attribute (attribute data type) on which to search, a search filter operator that will determine what to match, and the value of the attribute being sought.

Steps

1. On the **LDAP Filter** window, enter a search filter in the text field.
2. Ensure the syntax and variable names are correct. For general information about search filters, consult your directory documentation.
3. Click **Next** to complete the configuration to query attributes from the directory server.

   Later in the workflow, you can use the attribute values returned from your directory server in the applicable contract fulfillment window, the issuance criteria window, or both, to fulfill your use case.

Example

Suppose you want to locate user records by matching the mail Active Directory (AD) user attribute against an extended attribute, eml, in your access token contract for the purpose of mapping attributes to an OpenID Connect policy. As a passed-in variable from the access token, `${eml}` is shown underneath the **Filter** text field.

On the **LDAP Filter** window, enter the following filter in the **Filter** text field.

```
mail=${eml}
```

mail

   An AD user attribute containing the email address of the user

${eml}

   The value of the extended attribute (eml) in the access token contract

> ⓘ **Important:**
>
> You must use the `${}` syntax to retrieve the value of the enclosed variable.

Specifying database tables and columns
In the **Database Table and Columns** window, you can specify exactly where additional data can be found to fulfill your use case. You can only use one table as a source of data for a database query.

About this task
For more information about each field, see the following table.

Field	Description
Schema	Lists the table structure that stores information within a database. Some databases require selection of a specific schema for database queries. Other databases do not require selection of a schema.
Table	Displays the tables contained in the database. Select the table to retrieve data from the datastore.
Columns to return from SELECT	Displays selected columns from the selected tables. Select the columns that are associated with the desired attributes you want to return from the database queries.

ⓘ **Important:**

For MySQL users — To allow for table and column names that might contain spaces, PingFederate Bridge inserts double quotes around the names at runtime. To avoid SQL syntax errors resulting from the quotes, add the session variable `sql_mode=ANSI_QUOTES` to the Java Database Connectivity (JDBC) connection string of your datastore instance, as in the following example.

```
jdbc:mysql://myhost.mydomain.com:3306/pf?sessionVariables=sql_mode=ANSI_QUOTES
```

Alternatively, you can configure the system variable *sql_mode* with the `ANSI_QUOTES` option. For more information, see *https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html*.

Steps

1. From the **Schema** list, choose a schema when applicable.
2. From the **Table** list, select a table.
3. Optional: To determine what attributes to examine, click **View Attribute Contract**.
4. Optional: To update an existing configuration where changes to the database might have occurred, click **Refresh**.
5. Under **Columns to return from SELECT**, choose a column name and click **Add Attribute**.

   ⓘ **Note:**

   You do not have to add a column here to use it as part of a search filter. Add only the column that is required by subsequent sibling configuration items, such as contract fulfillment or token authorization. Any added columns left unused are removed when the configuration is saved.

   Repeat this step to add more columns as needed.

Example

Suppose you, the identity provider (IdP), have a data table named **ACCESSTABLE** with thee columns: userid, department, and accesslevel. Your use case requires you to map accesslevel into a SAML contract to an SP.

On the **Database Table and Columns** window, select the **ACCESSTABLE** table and add the accesslevel column.

Reviewing datastore query configurations
You can review the configuration on the **Summary** window to determine whether to change your configuration, save those changes, or discard them.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
- To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

- To discard your changes, click **Cancel**.

Configuring IdP adapter contract fulfillment
You can map values into the IdP adapter contract using the **Adapter Contract Fulfillment** tab.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**
2. Click the **Instance Name** of the existing IdP adapter instance you want to configure.
3. Go to the **Adapter Contract Mapping** tab.

> ⓘ **Note:**
>
> If this is a child instance, select the **Override Adapter Contract** check box to modify the configuration unless you have already selected the override option in the **Extended Contract** tab, in which case the **Override Adapter Contract** check box is automatically selected for you.

4. Click **Configure Adapter Contract**.
5. Select a source from the **Source** list and specify a **Value** to associate with it.

   For more information about the **Source** list and the possible **Values**, see the following table.

Source	Description
Adapter	Select **Adapter** to use the attribute value returned by the IdP adapter without customization.
Context	Select **Context** to return specific information from the request.
JDBC, LDAP, or other types of datastores (if configured)	Select an attribute source when PingFederate Bridge should retrieve attribute value from a datastore.
	When you make this selection, the **Value** list is populated with attributes from your database, directory, or other datastore.
	Applicable only if you have added at least one attribute source on the **Attribute Sources & User Lookup** tab. For more information, see *Defining attribute sources and user lookup* on page 27.

Source	Description
Expression (if enabled)	Select **Expression** to support complex mapping requirements; for example, transforming incoming values into different formats. Additionally, the HTTP request is retrieved as a Java object rather than text. For this reason, select **Expression** as the source and use OGNL expressions to evaluate and return specific information from the HTTP request.
	Applicable only if you have enabled the use of expressions in PingFederate Bridge. For more information, see *Attribute mapping expressions* on page 37.
No Mapping	Select **No Mapping** to ignore the **Value** field.
Text	Select **Text** to return the value you enter under **Value**.
	There are a variety of reasons to use a static text value. For example, if the target web application provides a service based on the name of your organization, you may provide the attribute value as a constant.
	You can mix text with references to attributes from the IdP adapter contract by using the `${attribute}` syntax.
	You can also enter references to attributes from configured attribute sources by using the `${ds.attr-source-id.attribute}` syntax, where `attr-source-id` is the **Attribute Source ID** value you entered on **Attribute Sources & User Lookup**# **Data Store** and `attribute` is an attribute from the datastore. For more information, see *Defining attribute sources and user lookup* on page 27

6. Repeat these steps until all attributes are configured.

7. Click **Done**.

Defining issuance criteria for IdP adapter contract
You can manage criteria that PingFederate Bridge can evaluate to determine whether to issue an identity provider (IdP) adapter contract token for a user.

About this task

On the **Issuance Criteria** tab, define the criteria to satisfy in order for PingFederate Bridge to further process a request. Use this token authorization feature to conditionally approve or reject requests based on individual attributes.

Begin this optional configuration by choosing the source that contains the attribute to verify. Some sources are common to almost all use cases, such as **Mapped Attributes**. Other sources depend on the type of configuration, such as **JDBC**. Irrelevant sources are automatically hidden. Once you select a source, choose the attribute to verify. Depending on the selected source, the available attributes or properties vary. Specify the comparison condition and the desired value to compare to.

You can define multiple criteria, which must all be satisfied in order for PingFederate Bridge to move a request to the next phase. A criterion is satisfied when the runtime value of the selected attribute matches or does not match the specified value, depending on the chosen comparison method. The **multi-value contains ...** or **multi-value does not contain ...** comparison methods are intended for attributes that can contain multiple values. Such a criterion is considered satisfied if one of the multiple values match or does not match the specified value. Values are compared verbatim. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions.

ⓘ **Note:**

All criteria defined must be satisfied, or evaluated as true, for a request to move forward, regardless of how the criteria were defined. As soon as one criterion fails, PingFederate Bridge rejects the request and returns an error message.

Steps

1. Go to **Authentication**# **Integration**# **IdP Adapters**.
2. Click the name of the existing instance you want to change from the **Instance Name** list.
3. Click **Adapter Contract Mapping**# **Configure Adapter Contract**# **Issuance Criteria**.
4. From the **Source** list, select the attribute's source.

   Depending on the selection, the **Attribute Name** list populates with associated attributes. See the following table for more information.

Source	Description
Adapter	Select to evaluate attributes from the IdP adapter instance.
Context	Select to evaluate properties returned from the context of the transaction at runtime.
	ⓘ **Note:** As the **HTTP Request** context value is retrieved as a Java object rather than text, attribute mapping expressions are more appropriate to evaluate and return values.
JDBC, **LDAP**, or other types of datastore (if configured)	Select to evaluate attributes returned from a data source.
Mapped Attributes	Select to evaluate the mapped attributes.

5. From the **Attribute Name** list, select the attribute to be evaluated.
6. From the **Condition** list, select the comparison method.

   Available methods:

   - **equal to**
   - **equal to (case insensitive)**
   - **equal to DN**
   - **not equal to**
   - **not equal to (case insensitive)**
   - **not equal to DN**
   - **multi-value contains**
   - **multi-value contains (case insensitive)**
   - **multi-value contains DN**
   - **multi-value does not contain**
   - **multi-value does not contain (case insensitive)**
   - **multi-value does not contain DN**

   ⓘ **Note:**

   The first six conditions are intended for single-value attributes. Use one of the **multi-value ...** conditions for PingFederate Bridge to validate whether one of the attribute values matches the specified value. When an attribute has multiple values, using a single-value condition causes the criteria to fail.

7. In the **Value** field, enter the comparison value.

> ⓘ **Note:**
>
> Values are compared verbatim. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions. For more information, see *Attribute mapping expressions* on page 37.

8. In the **Error Result** field, enter a custom error message.

   To use localized descriptions, enter a unique alias in the **Error Result** field, such as `someIssuanceCriterionFailed`. Insert the same alias with the desired localized text in the applicable language resource files, located in the `<pf_install>`/pingfederate/server/default/conf/language-packs directory.

   If not defined, PingFederate Bridge returns `ACCESS_DENIED` when the criterion fails at runtime.

9. Click **Add**.

10. Optional: Repeat to add more criteria.

11. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions. For more information, see *Attribute mapping expressions* on page 37.

    a. Click **Show Advanced Criteria**.
    b. In the **Expression** field, enter the required expressions.
    c. Optional: In the **Error Result** field, enter an error code or message.

    > ⓘ **Note:**
    >
    > If the expressions resolve to a string value instead of `true` or `false`, the returned value overrides the **Error Result** field value.

    d. Click **Add**.
    e. Optional: Click **Test**, enter values in the applicable fields, and verify the results.
    f. Optional: Repeat to add multiple criteria using attribute mapping expressions.

Reviewing an IdP adapter contract

From the **Summary** tab in the Adapter Contract Mapping workflow, you can review an adapter contract and make changes as needed.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
- To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

- To discard your changes, click **Cancel**.

Reviewing and saving an IdP adapter configuration

Review your IdP configuration and save your changes on the **Summary** tab.

Steps

- To keep your changes, click **Save**.

- To amend your configuration, click the name of the corresponding tab and then follow the configuration wizard to complete the task.
- To discard your changes, click **Cancel**.

Attribute mapping expressions
As of PingFederate Bridge 10.1, the use of expressions is enabled by default. You can use the Expression Admin administrative role to map user attributes by using OGNL expressions.

If you upgraded to PingFederate Bridge 10.1 from a previous version, the use of expressions is still enabled or disabled based on the configuration in the earlier version. Also, when upgrading PingFederate Bridge to 10.1 or later, administrative users who were granted the Admin role in the earlier installation are granted the Expression Admin role automatically.

Managing authentication applications

You can create and manage authentication applications that use the Authentication API.

About this task

Authentication applications display user interfaces to collect credentials when authentication is completed through the PingFederate Bridge authentication API. The default authentication application is used for authentication sources that support the authentication API functionality and are invoked directly, rather than as part of an authentication policy.

Steps

1. To manage authentication applications, go to **Authentication**# **Integration**# **Authentication API Applications**.
2. To toggle the availability of authentication API support, select or clear the **Enable Authentication API** check box.

   This check box is not selected by default.
3. To toggle the availability of the Authentication API Explorer, select or clear the **Enable API Explorer** check box.

   Applicable and shown if the **Enable Authentication API** check box is selected.

   When shown, this check box is selected by default.
4. Under the **Default Authentication Application** section, perform any of the following actions.

Option	Action
Default Authentication Application	Select an application from the drop-down to designate as the default authentication application.
Check Usage	Click to open a pop-up window listing the configurations in which the authentication is used. Only available for the default authentication application.
Add Authentication Application	Click to add a new authentication. See *Configuring authentication applications* on page 38.
Delete/Undelete	Click to remove an authentication application or cancel the removal request.

5. To save your changes, click **Save**.

Configuring authentication applications

You can create or configure an authentication application that employs the Authentication API.

Steps

1. Go to **Authentication**# **Integration**# **Authentication API Applications**.

    ▪ To create a new application, click **Add Authentication Application**.
    ▪ To modify an existing application, click the **Application Name**.

2. In the **Authentication Application** window, provide information for each field.

    For more information, see the following table.

Field	Description
Name	The name of the authentication application.
Description	An optional description of the authentication application.
URL	The URL of the authentication application.
Additional Allowed Origins	Enter any number of trusted origins to enable cross-origin resource sharing (CORS) support for the authentication API endpoint.

Once configured, client-side web applications from the trusted origins are allowed to make requests to the authentication API endpoint. For more information about CORS, see *W3C's recommendation on Cross-Origin Resource Sharing*.

Sample entry	Expected behavior
`https:// www.example.com`	CORS requests originating from https://www.example.com are allowed.
`https:// www.example.com:8080`	CORS requests originating from https://www.example.com:8080 are allowed.
`https:// www.example.com:*`	CORS requests originating from https://www.example.com:<*any port*> are allowed.

> ⓘ **Note:**
>
> Given this sample entry, a port number is required in the Origin request header.

> ⓘ **Important:**
>
> While using the wildcard character provides the convenience of allowing multiple origins with one entry, consider adding individual origins to limit CORS requests to a list of trusted hosts.

Click **Add** to save an entered origin. Repeat to add multiple origins.

3. To keep your configuration, click **Save** to or click **Cancel** to discard any changes made.

Configuring a default URL and error message

As an identity provider (IdP), you can optionally prompt end users to confirm their single logout (SLO) requests and provide a default URL indicating a successful SLO to the end-user, if no other page is designated.

About this task

You can also customize an error message to be displayed as part of the error page rendered in the end-user's browser if an error occurs during IdP-initiated single sign-on (SSO). For example, you might consider modifying the default text to include useful information regarding whom the user should contact or what their next step should be.

Your application or your partner's application can supply the SLO URL at runtime. However, if none is provided, PingFederate Bridge will use the default value you enter on this window. For more information, see *IdP endpoints*.

If you leave the default URL blank, PingFederate Bridge provides a built-in landing page for the user. This web page is among the templates you can modify with your own branding or other information. For more information, see *Customizable user-facing pages*).

Steps

1. Go to **Authentication**# **Integration**# **IdP Default URL**.
2. Select the check box to prompt the user to confirm SLO.
3. Enter a default URL to send the user to on successful SLO.
4. Enter a custom error message to display on unsuccessful SLO.

   > ⓘ **Note:**
   >
   > The error message is displayed only when the application calling the start-SSO endpoint does not explicitly provide its own error page URL. The default entry in this field is used to localize the message. For information about how to use the PingFederate Bridge localization feature, see *Localizing messages for end users*. If localization is not needed, you can also specify a default message in this field.

5. Click **Save** to save your changes.

Policies

You can configure policies, fragments, selectors, policy contracts, and sessions under **Authentication**# **Policies**.

An authentication policy is a tree of authentication sources, selector instances, or a combination of them, that defines the decision to route a request through a series of approved authentication sources with an optional authentication policy contract or a local identity profile at the end or to deny the request.

Administrators can enable authentication policies on identity provider (IdP) browser single sign-on (SSO) requests, adapter-to-adapter requests, and browser-based OAuth authorization code and implicit flows. Administrators can also enable authentication policies on service provider (SP)-initiated Browser SSO requests received at the /sp/startSSO.ping endpoint. Individual policies can be disabled, as needed.

The order of authentication policies matters because the policy engine starts from the first policy and works its way down. At runtime, the policy engine derives an authentication tree from the applicable policies and either approves or denies a request.

Policy paths, authentication policy contracts, and local identity profiles

Policy paths

An authentication policy starts with either a selector instance or an authentication source. Authentication sources and most selectors have two results, **Success** or **Fail**, **Yes** or **No**. Each result forms a policy path.

A policy path is open-ended if it contains only one or more selector instances, without any authentication sources. In this scenario, the policy engine continues to the next applicable authentication policy, if any.

A policy path is closed-ended if it contains one or more authentication sources, with or without any selector instances. A closed-ended path can optionally end with an authentication policy contract or a local identity profile.

> ⓘ **Note:**
>
> A policy path is also closed-ended if it ends with an instance of a custom authentication selector that returns an IdP adapter instance ID or the connection ID of an IdP connection. Because the custom selector returns an authentication source, such a closed-ended path cannot end with an authentication policy contract or a local identity profile. Instead, it must end with an action of **Done** or **Restart**.

Authentication policy contracts and local identity profiles

An authentication policy contract can harness attribute values obtained from all authentication sources along the path leading up to it. Administrators can select the same authentication policy contract or local identity profile for different closed-ended paths, in one or more authentication policies, and fulfill them differently to suit the requirements. To enforce the same set of authentication policies in multiple use cases, map the authentication policy contract to the applicable Browser SSO connections and OAuth grant-mapping configuration.

A policy becomes more complex as the number of paths grows with the number of authentication sources and selector instances.

Multiple policies and runtime behavior

A complex policy can cover a lot of ground. However, depending on the authentication requirements, administrators can also create multiple policies to suit their needs.

When a request arrives at PingFederate Bridge, the policy engine skips all disabled policies and any closed-ended paths that are inapplicable to the request. A closed-ended path is considered inapplicable to a request in any of the following conditions:

▪ The local identity profile at the end of a path is associated with an authentication policy contract that is not mapped to the invoking use case or is blocked by the virtual server ID included in the request.
▪ The authentication policy contract at the end of a path is not mapped to the invoking use case or is blocked by the virtual server ID included in the request.
▪ The last authentication source at the end of a path, that does not end with an authentication policy contract or a local identity profile, is not mapped to the invoking use case or is blocked by the virtual server ID included in the request.

> ⓘ **Note:**
>
> Virtual server IDs are not applicable to adapter-to-adapter mappings or OAuth use cases.

After pruning inapplicable policies and paths, the policy engine starts evaluating the request against the first applicable policy. Generally speaking, the policy engine moves on to the next applicable policy when it hits the end of an open-ended path, as indicated by an action of **Continue**, and stops when it hits the end of a closed-ended path, as indicated by an authentication policy contract or an action of **Done** or **Restart**. Depending on the policies, the policy engine might find an authentication source, a series of authentication sources, or no authentication source at all.

Default authentication sources

In the event that a request has only passed through an open-ended path and the policy engine finds no authentication source after evaluating the request through all the applicable policies, it picks the first applicable default authentication source. A default authentication source is considered applicable if it is mapped to the use case of the request.

If the policy engine cannot find a default authentication source and the **Fail if policy engine finds no authentication source** check box is not selected, PingFederate Bridge chooses an authentication source based on the following prioritized preferences:

1. If the request comes with an IdpAdapterId query parameter or a pfidpaid cookie, and if the authentication source specified by the query parameter or the cookie is mapped to the corresponding use case, PingFederate Bridge uses the specified authentication source. If the authentication source is not mapped, PingFederate Bridge denies the request and returns an error message.

   > ⓘ **Note:**
   >
   > If the request presents both the IdpAdapterId query parameter and the pfidpaid cookie, the IdpAdapterId query parameter takes precedence.

2. If the request comes with neither an IdpAdapterId query parameter nor a pfidpaid cookie, and if there is only one authentication source mapping, PingFederate Bridge uses the mapped authentication source.

   > ⓘ **Note:**
   >
   > If there are multiple authentication-source mappings, PingFederate Bridge returns the available authentication sources and lets the user authenticate through one of them. If the user selected the **Remember selection** check box and successfully authenticated, PingFederate Bridge returns a pfidpaid persistent cookie, identifying the user's preference.

If the **Fail if policy engine finds no authentication source** check box is selected, PingFederate Bridge denies the request and returns an error message.

> ⓘ **Note:**
>
> If a request has passed through a closed-ended path, the policy engine has already found at least one authentication source for the user; in this scenario the policy engine ignores all default authentication sources.

Tracked HTTP request parameters

The policy engine is capable of tracking HTTP request parameters that it receives from the initial request and making them available to authentication sources, selector instances, and contract mappings throughout the policy.

Local identity profiles and authentication policy contracts

PingFederate Bridge empowers administrators to deliver a secure and easy-to-use customer authentication, registration, and profile management solution. A typical use case involves an HTML Form

Adapter instance, a local identity profile, an authentication policy contract, and an IdP authentication policy. The HTML Form Adapter captures user attributes and maps them into an authentication policy contract through a local identity profile. In terms of configuration, the latter is accomplished by placing a local identity profile at the end of a policy path and completing the **Local Identity Mapping**# **Contract Fulfillment** configuration.

Policy enforcement based on OAuth clients

PingFederate Bridge can enforce authentication polices based on the requesting OAuth client, among other factors. To do that, you can include an OAuth Client Set Authentication Selector or an Extended Property Authentication Selector in a policy. For more information, see *Configuring the OAuth Client Set Authentication Selector* on page 55 and *Configuring the Extended Property Authentication Selector* on page 49

Another option is to leverage the policy enforcement rule that PingFederate Bridge ignores authentication policy contract branches that are not mapped to an access token manager (ATM) that the requesting client can access. This rule also applies to policy branches that end on an authentication source. For more information, see *Managing access token mappings*.

Keep that policy enforcement rule in mind when you configure clients and ATMs. You can configure a client so that it can access only its default ATM. You can also configure an ATM so that only specific clients can access it. Those settings affect the authentication policy. For more information, see *Configuring OAuth clients* and *Access token management*.

Defining authentication policies

You can manage authentication policies and settings on the **Policies** window.

Steps

1. Go to **Authentication**# **Policies**# **Policies**.
2. On the **Policies** tab, select the **IdP Authentication Policies** check box if you want to enable authentication policies for identity provider (IdP) browser single sign-on (SSO) requests, adapter-to-adapter requests, and browser-based OAuth authorization code and implicit flows.
3. Select the **SP Authentication Policies** check box if you want to enable authentication policies for service provider (SP)-initiated browser SSO requests received at the `/sp/startSSO.ping` endpoint.

   > ⓘ **Note:**
   >
   > Selecting the **SP Authentication Policies** check box does not enable authentication policies for IdP browser SSO requests, adapter-to-adapter requests, and browser-based OAuth authorization code and implicit flows.

4. Select the **Fail if policy engine finds no authentication source** check box if you want PingFederate Bridge to deny the requests and to return an error message when the policy engine finds no authentication source or authentication policy contract from the applicable policies and none of the default authentication sources are applicable. This check box is not selected by default.

5. On the **Policies** window, click **Add Policy** to create an authentication policy.

> ⓘ **Tip:**
>
> If you want to create a new policy based on an existing policy, select the **Copy** action.

  a. Enter a name and, optionally, a description of the policy.
  b. From the **Policy** list, choose an authentication source, an IdP adapter instance or an IdP connection, a selector instance, or a fragment.

  > ⓘ **Note:**
  >
  > If you start this new policy by copying an existing policy, your new policy is pre-populated. Modify the policy to suit your new use cases.

  > ⓘ **Tip:**
  >
  > When implementing your authentication requirements, think of authentication sources and selectors as checkpoints.

  Options

  > For the PingID Adapter, IdP adapters developed using the `IdpAuthenticationAdapterV2` interface from the PingFederate SDK, including the HTML Form Adapter, and SAML 2.0 IdP connections supporting the SP-initiated browser SSO profile, you can specify a user ID to be passed in from an earlier-factor adapter.
  >
  > Click **Options** and follow the on-window instructions to select the source and the attribute to be used as the incoming user ID.

  > ⓘ **Note:**
  >
  > The **User ID Authenticated** check box must be selected for users to register as a new PingID user. Otherwise, the adapter automatically fails.

  Rules

  > For any authentication source, you can optionally create one or more rules to define additional successful results. For example, if you want to deploy multifactor authentication using the PingID Adapter in stages by groups, you can create a rule to check for group membership information and only apply the PingID authentication flow to users who are members of certain groups.
  >
  > Click **Rules** and follow the on-window instructions to manage your rules.

  All results, including those based on rules, are displayed under the selected authentication source or selector instance. Each result forms a policy path.
  c. For each policy path, select a policy action from the list.

  > ⓘ **Important:**
  >
  > As of PingFederate 10.2, you can select **Fragments** as the policy action and then select a policy fragment that you have created. (See *Policy fragments* on page 60 for information about creating

policy fragments.) When you select a fragment, click **Fragment Mapping** and use the in-product help links to access the topics that describe how to configure the mapping.

- If additional processing is required, repeat *step 4b*.
- If the policy path is extended from an authentication source and it is the end of the path, select **Done** or **Restart**, which marks this policy path a closed-ended path.

> ⓘ **Tip:**
>
> A policy path is closed-ended if it contains one or more authentication sources, with or without any selector instances. A closed-ended path can optionally end with an authentication policy contract or a local identity profile.

If you need to reuse an authentication policy in multiple use cases, select an authentication policy contract or a local identity profile as the last policy action of a path, configure its contract fulfillment, and map the authentication policy contract to the applicable Browser SSO connections or OAuth grant-mapping configuration. Click **... Mapping** underneath your selection and then follow the on-window instructions to complete the contract fulfillment configuration.

> ⓘ **Note:**
>
> A policy path is also closed-ended if it ends with an instance of a custom authentication selector that returns an IdP adapter instance ID or the connection ID of an IdP connection. Because the custom selector returns an authentication source, such a closed-ended path cannot end with an authentication policy contract or a local identity profile. Instead, it must end with an action of **Done** or **Restart**.

The **Restart** policy action provides users the opportunity to do over. When triggered, the policy engine routes the requests back to the first checkpoint of the invoked authentication policy. It makes most sense to use the **Restart** policy action for a **Fail** policy path if the policy engine can route the request differently based on user input prompted by an authentication source. For a sample use case, see step 5 in *Enabling third-party identity providers*.

> ⓘ **Note:**
>
> Undesirable looping behaviors can occur if you select **Restart** for the **Fail** path at the root of an authentication policy tree. PingFederate Bridge mitigates this risk by automatically limiting the number of policy restarts per transaction.

- If the policy path is extended from a selector instance and it is the end of the path without any prior authentication source, select **Continue**, which leaves this path as an open-ended path.

> ⓘ **Tip:**
>
> A policy path is open-ended if it contains only selector instances without any authentication sources. In this scenario, the policy engine continues to the next applicable authentication policy, if any.

d. Click **Done** to go back to the **Policies** window.

Your policy is enabled by default. As needed, toggle its status to disable the policy.

6. Optional: Repeat *step 4* to create additional authentication policies.

> ⓘ **Important:**
>
> The order of authentication policies matters because the policy engine starts from the first policy and works its way down. As needed, reorder your policies by using the up and down arrows.

7. If any individual policy is no longer required, select the **Delete** action or toggle its status to disable the policy.

8. Optional: On the **Policies**# **Default Authentication Sources** tab, select one or more default authentication sources from the list for the policy engine to fall back on when it finds no authentication source from the applicable policies.

> ⓘ **Important:**
>
> Order matters because the policy engine starts from the first default authentication source on the list and works its way down. As needed, reorder your authentication sources by using the up and down arrows. There is no default selection.

9. Optional: On the **Policies**# **Tracked HTTP Parameters** tab, add one or more HTTP request parameters to be tracked throughout a request.

> ⓘ **Important:**
>
> For each instance of the HTTP Request Parameter Authentication Selector that you place in a policy as the second, or subsequent, checkpoint, add its configured **HTTP Request Parameter Name** value here. By doing so, the policy engine preserves the parameter it receives from the initial request and makes it available to the selector instance throughout the policy. For more information, see *Configuring the HTTP Request Parameter Authentication Selector* on page 53.

10. Click **Save**.

Managing authentication selector instances

You can manage authentication selectors on the **Selectors** window in the PingFederate Bridge administrative console.

Steps

- Go to **Authentication**# **Policies**# **Selectors**.
- To configure a new instance, on the **Selectors** window, click **Create New Instance**.
- To modify an existing instance, select it by its name under **Instance Name**.
- To remove an existing instance or to cancel the removal request, click **Delete** or **Undelete**.

> ⓘ **Note:**
>
> You can only remove a selector instance if it is not deployed in any authentication policy.

Choosing a selector type

Choose an authentication selector instance from the list of available selector types.

Steps

1. Go to **Authentication**# **Policies**# **Selectors**.
2. Click **Create New Instance**.
3. In the **Instance Name** field, enter an instance name.

4. In the **Instance ID** field, enter an instance ID.
5. From the **Type**list, select the desired type of authentication selector.

Configuring an authentication selector instance
The configuration of an authentication selector instance varies depending on the authentication selectors deployed on your server.

Steps

1. Refer to subsequent topics for configuration steps of each of the bundled authentication selectors.
2. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Configuring the CIDR Authentication Selector
The CIDR Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on the IP address of an incoming single sign-on request.

About this task

Use this selector in authentication policies to choose from authentication sources that share a similar level of assurance, such as among multiple HTML Form Adapter instances or between a Kerberos Adapter instance and an X.509 identity provider (IdP) Adapter instance. For example, use this selector in authentication policies to route internal requests to a Kerberos Adapter instance.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, click **Add a new row to 'Networks'** and enter a network range. Click **Update**.

> (i) **Note:**
>
> To see the **Add a new row to 'Networks'** option, ensure you have set the **Authentication Selector Instance** type to **CIDR Authentication Selector** on the **Type** tab.

Sample IPv4 network range

Enter `192.168.101.0/24` to cover 256 IPv4 addresses, ranging from `192.168.101.0` through `192.168.101.255`.

Sample IPv6 network range

Enter `2001:db8::/123` to cover 32 IPv6 addresses, ranging from `2001:db8::` through `2001:db8::1f`.

5. Optional: Repeat the previous step to add more network ranges.

> ⓘ **Note:** Display order does not matter.

> ⓘ **Tip:**
>
> If you want to include all IPv4 addresses for testing, add two separate ranges: `0.0.0.0/1` and `128.0.0.0/1`. The CIDR Authentication Selector interprets a specification of `0.0.0.0/0` as an empty range rather than as a wildcard for all addresses.

Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.

6. Optional: Enter a **Result Attribute Name** value.

> ⓘ **Note:**
>
> This field provides a means to indicate in the SAML assertion whether a network range was matched during processing; the value is either `Yes` or `No`. Any authentication sources configured as a result of this authentication selector must have their attribute contract extended with the value of the **Result Attribute Name** field in order to use its value to fulfill an attribute contract or for issuance criteria.

7. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Results

When you place this selector instance as a checkpoint in an authentication policy, it forms two policy paths: **Yes** and **No**. If the IP address of an incoming single sign-on (SSO) request matches one of the defined network ranges, the selector returns true. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **Yes**. If the IP address of an incoming SSO request matches none of the defined network ranges, the selector returns false. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **No**.

Configuring the Cluster Node Authentication Selector
The Cluster Node Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on the PingFederate Bridge cluster node that is servicing the request in authentication policies.

About this task

For example, this selector allows you to choose whether Integrated Windows Authentication (IWA) is attempted based on the PingFederate Bridge cluster node with which a Key Distribution Center (KDC) is associated.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.

2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.

3. On the **Type** tab, configure the basics of this authentication selector instance.

4. On the **Authentication Selector** window, select the **Field Value** on which to branch policy paths. The authentication selector provides a means of choosing authentication sources at runtime based on the cluster node on which it is executing.

> **Node Index**
>
> > Select **Node Index** to use the `pf.cluster.node.index` value specified in `run.properties`.
>
> **Node Tag**
>
> > Select **Node Tag** to use the `node.tags` values specified in `run.properties`.

5. On the **Selector Result Values** window, specify the relevant node index or node tag values.

> ⓘ **Note:**
>
> Each selector result value forms a policy path when you place this selector instance as a checkpoint in an authentication policy.

> a. In the **Result Values** field, enter a node index or node tag value based on your cluster configuration and click **Add**. This value should correspond to a node index or node tag of one of the engine nodes in the cluster.
> b. Optional: Add more values to differentiate criteria for authentication selection.

> ⓘ **Note:**
>
> Display order does not matter.
>
> Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Click **Delete** to remove an entry.

6. Complete the configuration.
> a. On the **Summary** tab, click **Done**.
> b. On the **Selectors** window, click **Save**.

Configuring the Connection Set Authentication Selector

The Connection Set Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on a match found between the target service provider (SP) connection used in a single sign-on (SSO) request and SP connections configured within PingFederate Bridge.

About this task

This selector allows you to override connection authentication selection on an individual connection basis in one or more authentication policies.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. From the **Type** list, make sure you select **Connection Set Authentication Selector**.
5. Click **Next**. In the **Authentication Selector** window, click **Add a new row to 'Connections'**.
6. From the **Connection** list, select an SP connection and click **Update**.

7. Optional: Repeat the previous step to add more connections. Display order does not matter.

   Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.

8. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Results

When you place this selector instance as a checkpoint in an authentication policy, it forms two **Yes** and **No** policy paths. If the invoking SP connection matches one of the connections from the set, the selector returns true. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **Yes**. If the invoking SP connection matches none of the connections from the set, the selector returns false. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **No**.

Configuring the Extended Property Authentication Selector

The Extended Property Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on a match found between a selector result value and an extended property value from the invoking browser-based SSO connections or OAuth client.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, select a property from the **Extended Property** list.

> (i) **Note:**
>
> The extended property is the property that this selector instance should look for from the invoking connection or client, and compare the populated property value, or values if it is a multivalued extended property, against the selector result values defined in this selector instance.

5. On the **Selector Result Values** tab, specify one or more expected result values.
   a. Enter the exact, case-sensitive, value under **Result Values** and click **Add**.
   b. Optional: Add more values to differentiate criteria for authentication selection.

      Display order might matter.

      Expected result values are always sorted alphabetically in ascending order here.

      When you place this selector instance as a checkpoint in an authentication policy, each selector result value forms a policy path. The display order of the resulting policy paths matches the display order here, which may impact the policy outcome. When the policy engine reaches this selector instance, the selector starts from top to bottom. As soon as it finds a match, it exits and returns true. The matching mechanism varies, depending on the type of the extended property selected in step 4.

      Matching mechanism for single-value extended properties

         The selector compares the property value populated in the invoking connection or client against the configured selector result value. When multiple selector result values exist, the selector starts from the top. If the current selector result value is a case-sensitive exact match, it returns true and exits. Otherwise, it moves on to the next selector result value and tries again.

For example, assume this selector instance, named ExtProps, is configured with expected result values of `Alpha`, `Bravo`, and `Charlie`. The invoking connection is populated with an extended property value of `Bravo`, and this selector instance is placed as a checkpoint in an authentication policy as follows.

```
ExtProps
+--Alpha
|   <policy path>
|
+--Bravo
|   <policy path>
|
+--Charlie
    <policy path>
```

Given this setup, the selector returns true and exits when it reaches the second selector result value. The policy engine regains control of the request and proceeds with the policy path configured for the selector result value of `Bravo`.

Matching mechanism for multivalued extended properties

The selector compares the property values populated in the invoking connection or client against the configured selector result value. If any one of the property values from the invoking connection or client is a case-sensitive exact match, the selector returns true and exits. When multiple selector result values exist, the selector starts from the top. If the current selector result value is a case-sensitive exact match to any one of the property values from the invoking connection or client, it returns true and exits. Otherwise, it moves on to the next selector result value and tries again.

For example, assume the previous selector instance remains. The invoking connection is populated with extended property values of `Alpha` and `Charlie`, and this selector instance remains as a checkpoint in an authentication policy.

In this scenario, the selector returns true and exits when it reaches the first selector result value. The policy engine regains control of the request and proceeds with the policy path configured for the selector result value of `Alpha`. Even though `Charlie`, the expected selector result value, is also a case-sensitive exact match to `Charlie`, one of the property values from the invoking connection, because the selector has already exited and returned control to the policy engine when it reaches `Alpha`, the policy engine will never execute the policy path configured for the selector result value of `Charlie`.

Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Click **Delete** to remove an entry.

6. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Example

1. Go to **System**# **Server**# **Extended Properties**.
2. On the **Extended Properties**window, define a multivalued extended property, and name it configStatus.
3. Create an SP connection with the following characteristics:

   ▪ On the **Extended Properties** window, add two values for the configStatus extended property: `DEV` and `TEST`.
   ▪ On the **Attribute Source Mapping** window, map an authentication policy contract to the service provider (SP) connection. The policy contract name is `APC`.

4. Create an instance of the Extended Property Authentication Selector with the following characteristics:
   - On the **Type** tab, name the selector instance `ExProps`.
   - On the **Authentication Selector** tab, select configStatus from the list.
   - On the **Selector Result Values** tab, enter `DEV` and `TEST`.
5. Create and activate the following identity provider (IdP) authentication policy.

```
ExtProps
+--DEV
|  OpenToken
|  +--Fail: Done
|  +--Success: APC
|
+--TEST
    HTML
    +--Fail: Done
    +--Success: APC
```

Configure each `APC` to fulfill values obtained from its preceding adapter instance.

When processing SSO requests intended for this SP connection, because the policy engine is able to match one of the populated property values, `DEV`, from the SP connection to the first selector result value, also `DEV`, it will always invoke the OpenToken IdP Adapter instance based on the `DEV` policy path. The `TEST` policy path is never executed for this SP connection.

On the other hand, if you remove `DEV`, an extended property value, from the SP connection, the policy engine will route SSO requests intended for this SP connection to the HTML Form Adapter instance based on the `TEST` policy path. The `DEV` policy path is never executed for this SP connection.

Configuring the HTTP Header Authentication Selector
The HTTP Header Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on a match found in a specified HTTP header.

About this task

Use this selector in one or more authentication policies to choose from authentication sources that share a similar level of assurance, such as among multiple HTML Form Adapters or between a Kerberos Adapter and an X.509 Adapter. For example, use this selector to choose an authentication source based on the user's browser identified by the User-Agent HTTP header.

ⓘ **Important:**

Do not use this selector to determine whether an authentication source with a higher level of assurance should be bypassed because HTTP request headers could potentially be forged.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, click **Add a new row to 'Results'**.

5. Enter an expression for use when inspecting the HTTP header value of the target HTTP header under **Match Expression**, and click **Update**.

> ⓘ **Note:**
>
> Wildcard entries are allowed, such as `*value*`.

6. Optional: Repeat the previous step to add more expressions. Display order does not matter.

> ⓘ **Note:**
>
> Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.

7. In the **Header Name** field, enter the type of HTTP header you want the selector to inspect. This field is not case-sensitive.
8. Optional: To disable case-sensitive matching between the HTTP header values from the requests and the **Match Expression** values specified on this window, clear the **Case-Sensitive Matching** check box.

   The **Case-Sensitive Matching** check box is selected by default.
9. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Results

When you place this selector instance as a checkpoint in an authentication policy, it forms two policy paths: **Yes** and **No**. If the value of the specified HTTP header matches one of the configured values, the selector returns true. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **Yes**. If the value of the specified HTTP header matches none of the configured values, the selector returns false. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **No**.

Example

To detect the most common browsers based on the User-Agent HTTP request header, configure an HTTP Header Authentication Selector instance as follows.

1. Enter these entries under **Match Expression**.

Browser	Expression
Chrome	`*Chrome*`
Firefox	`*Firefox*`
Internet Explorer	`*MSIE*`

> ⓘ **Tip:**
>
> For more information, see *User-agent string changes* from Microsoft.

Safari	`*Safari*`

2. In the **Header Name** field, enter `User-Agent`.

Configuring the HTTP Request Parameter Authentication Selector
The HTTP Request Parameter Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on query parameter values.

About this task

Use this selector in one or more authentication policies to choose from authentication sources that share a similar level of assurance, such as among multiple instances of the HTML Form Adapter or between a Kerberos Adapter instance and an X.509 Adapter instance. For example, use an instance of this selector to choose an authentication experience based on the reward program information indicated by a query parameter in the single sign-on (SSO) request.

> ⓘ **Important:**
>
> Do not use this selector to determine whether an authentication source with a higher level of assurance should be bypassed because query parameters could potentially be forged.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, configure the applicable selector instance settings.
   a. Enter the exact, case-sensitive name of the request parameter in the **HTTP Request Parameter Name** field.

   > ⓘ **Important:**
   >
   > The policy engine is capable of tracking HTTP request parameters that it receives from the initial request and making them available to selector instances throughout the policy. If you plan on using this selector instance as the second, or subsequent, checkpoint in at least one authentication policy, add the **HTTP Request Parameter Name** value on the **Tracked HTTP Parameters** window. For more information, see *Defining authentication policies* on page 42.

   b. Optional: To disable case-sensitive matching between the HTTP request parameter values from the requests and the **Match Expression** values specified on the **Selector Result Values** window, clear the **Case-Sensitive Matching** check box.

   > ⓘ **Note:**
   >
   > The **Case-Sensitive Matching** check box is selected by default.

   c. Optional: Enable policy paths to handle additional scenarios.
   For more information, see the following table.

Field	Description
Enable 'Any' Result Value	Each configured selector result value forms a separate authentication policy path.
	Select this check box if you want to enable a single policy path for the scenario where the HTTP request parameter value matches any one of the configured selector result values.
	This check box is not selected by default.

Field	Description
Enable 'No Match' Result Value	Selector evaluation fails and the next applicable authentication policy is executed when the HTTP request parameter value does not match any of the configured selector result values.
	Select this check box if you want to enable a policy path to handle this scenario.
	This check box is not selected by default.
Enable 'Not in Request' Result Value	Selector evaluation fails and the next applicable authentication policy is executed if the HTTP request parameter is not found.
	Select this check box if you want to enable a policy path to handle this scenario.
	This check box is not selected by default.

5. On the **Selector Result Values** window, enter a request parameter value under **Result value**, and then click **Add**.

   > ⓘ **Note:**
   >
   > Wildcard entries are allowed, such as `*value*`.

   > ⓘ **Important:**
   >
   > A more specific match is a better match and an exact match is the best match.

6. Optional: Repeat the previous step to add more request parameter values. Display order does not matter.

   > ⓘ **Note:**
   >
   > If you have not enabled the **Any** policy path in *step 4c*, each selector result value forms a policy path when you place this selector instance as a checkpoint in an authentication policy.
   >
   > If you have enabled the **Any** policy path, only one policy path is formed.
   >
   > Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Click **Delete** to remove an entry.

7. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Example

Suppose you enter three selector result values, `Central`, `Eastern`, and `Southern`, on the **Selector Result Values** window, as illustrated in the following screen capture.

If you have not enabled any additional policy paths in *step 4c*, as you place this selector instance as a checkpoint in an authentication policy, three policy paths are extended from the selector instance, one for each of the configured selector result values.

Configuring the OAuth Client Set Authentication Selector
The OAuth Client Set Authentication Selector selector allows you to override client authentication select on an individual client basis in one or more authentication policies.

About this task

The selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on a match found between the client information in an OAuth request and the OAuth clients configured in the PingFederate Bridge OAuth authorization server (AS).

> ⓘ **Note:**
>
> The OAuth Client Set Authentication Selector is only applicable to OAuth clients using the authorization code or implicit flow.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, click **Add a new row to 'Clients'**.

> ⓘ **Note:**
>
> If you do not see **Add a new row to 'Clients'**, go back to the **Type** tab and ensure you have selected **OAuth Client Set Authentication Selector** from the **Type** list.

5. From the **Client ID** list, select an OAuth client and click **Update**.
6. Optional: Repeat the previous step to add more clients.

    Display order does not matter.

    Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.
7. Complete the configuration.
    a. On the **Summary** tab, click **Done**.
    b. On the **Selectors** window, click **Save**.

Results

When you place this selector instance as a checkpoint in an authentication policy, it forms two policy paths: **Yes** and **No**. If the invoking client matches one of the clients from the set, the selector returns true. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **Yes**. If the invoking client matches none of the clients from the set, the selector returns false. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **No**.

Configuring the OAuth Scope Authentication Selector
The OAuth Scope Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors based on a match found between the scopes of an OAuth authorization request and scopes configured in the PingFederate Bridge OAuth authorization server (AS).

Before you begin

Go to **System**# **OAuth Settings**# **Authorization Server Settings** and configure one or more scopes.

About this task

This selector allows you to control the strength of authentication based on client access requirements. For example, if a client requires write access to a resource, you can deploy an instance of the OAuth Scope Authentication Selector in one or more authentication policies to choose an adapter that offers a stronger form of authentication, such as the X.509 client certificate, instead of username and password.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, select the required scopes, scope groups, or both.

> ⓘ **Note:**
>
> Both common and exclusive scopes are available for selection.

> ⓘ **Important:**
>
> This selector matches only scopes from OAuth authorization requests to the authorization endpoint, `/as/authorization.oauth2`. SAML single sign-on (SSO) requests do not match this authentication selector's criteria and result in a returned result value of `No`. If you are using this selector and selectors specific to SAML connections, list this selector first in the mapping list so that it takes precedence for OAuth without disrupting selector logic on SAML connections.

5. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Results

When you mark this selector instance as a checkpoint in an authentication policy, it forms two policy paths: **Yes** and **No**. If the requested scopes satisfy all the selected scopes, the selector returns true. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **Yes**. If the requested scopes do not satisfy all the selected scopes, the selector returns false. The policy engine regains control of the request and proceeds with the policy path configured for the result value of **No**.

Configuring the Requested AuthN Context Authentication Selector
The Requested AuthN Context Authentication Selector enables PingFederate Bridge to choose configured authentication sources or other selectors.

About this task

This selector chooses authentication sources or selectors based on the authentication contexts requested by a service provider (SP) for browser single sign-on (SSO) requests, or a relying party (RP) for OAuth with OpenID Connect use cases in authentication policies.

For browser SSO, this authentication selector works in conjunction with SP connections with SAML 2.0 only, using the SP-initiated SSO profile. Other browser SSO protocols do not support authentication context. For OAuth, clients supporting the OpenID Connect protocol must include the optional acr_values parameter in their authorization requests to indicate their preferred authentication context, or contexts.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** tab, configure the applicable selector instance settings.
   a. Select the **Add or Update AuthN Context Attribute** check box if you want to update the authentication context attribute value with the value specified in the **Selector Result Values** tab.

      When selected, which is the default, the check box on this window provides a means to:

      - Add the value of the authentication context determined by the selector into the SAML assertion.
      - When applicable, replace any value returned from the associated adapter instance with the selector-result value.
   b. Optional: Enable policy paths to handle additional scenarios.

      For more information, refer to the following table.

Field	Description
Enable 'No Match' Result Value	Selector evaluation fails and the next applicable authentication policy is executed if the requested authentication context does not match any of the configured selector result values.
	Select this check box if you want to enable a policy path to handle this scenario. This check box is not selected by default.
Enable 'Not in Request' Result Value	Selector evaluation fails and the next applicable authentication policy is executed if no requested authentication context is found.
	Select this check box if you want to enable a policy path to handle this scenario. This check box is not selected by default.

5. On the **Selector Result Values** window, specify the authentication contexts to use as the criteria.
   a. Enter the exact, case-sensitive parameter value under **Result Values**, and then click **Add**.

      ⓘ **Note:**

      The value can include URIs defined in *Authentication Context for the OASIS Security Assertion Markup Language (SAML) 2.0* or any other value agreed upon with the partner.

   b. Optional: Add more values to differentiate criteria for authentication selection.

      Display order does not matter.

      Each selector result value forms a policy path when you place this selector instance as a checkpoint in an authentication policy (regardless of whether you have enabled the **No Match** or **Not in Request** policy path in *step 4b*).

      Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Click **Delete** to remove an entry.
6. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Configuring the Session Authentication Selector

The Session Authentication Selector enables PingFederate Bridge to choose a policy path at runtime based on whether the user already has a PingFederate Bridge authentication session for a particular source.

Steps

1. Go to **Authentication**# **Policies**# **Selectors** to open the **Selectors** window.
2. On the **Selectors** window, click **Create New Instance** to start the **Create Authentication Selector Instance** workflow.
3. On the **Type** tab, configure the basics of this authentication selector instance.
4. On the **Authentication Selector** window, click **Add a new row to 'Authentication Sources'**.
5. Select an IdP adapter instance or an IdP connection from the list, enter a value under **Result Value** for the selected authentication source, then click **Update**.

   The **Result Value** field controls the label shown for the policy path created by the selected authentication source.

   > ⓘ **Note:**
   >
   > You must enable authentication sessions for the selected authentication source, or globally for all authentication sources, on the**Sessions** window. Click **Manage Sessions** to review and configure authentication sessions.

6. Optional: Repeat the previous step to add more authentication sources.

   Display order might matter.

   When you place this selector instance as a checkpoint in an authentication policy, each selector result value forms a policy path. The display order of the resulting policy paths matches the display order here, which may impact the policy outcome. When the policy engine reaches this selector instance, the selector starts from top to bottom. It exits and returns true as soon as it finds a match.

   As needed, use the up and down arrows to re-arrange the display order here, which also re-prioritizes the resulting policy paths.

   In addition, when no session exists for any of the defined sources, the result value for the first authentication source is returned unless the **Enable 'No Session' Result Value** check box is selected, in which case an additional policy path is added as the last path when this selector instance is placed as a checkpoint in an authentication policy.

   Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.

7. Optional: Select the **Enable 'No Session' Result Value** check box to create a separate policy path for the scenario where no session exists for any of the defined sources.

   This check box is not selected by default.

8. Complete the configuration.
   a. On the **Summary** tab, click **Done**.
   b. On the **Selectors** window, click **Save**.

Results

When you place this selector instance as a checkpoint in an authentication policy, each selector result value forms a policy path that you can define the desired authentication experience and requirements.

Example

The following screen capture illustrates a configuration where three authentication sources are defined and the **Enable 'No Session' Result Value** check box is selected.

When this selector instance (named Intranet sessions) is placed in a policy, four policy paths are formed.

Managing policy contracts

You can manage authentication policy contracts in the **Authentication**# **Policies**# **Policy Contracts** window.

Steps

1. Click **Create New Contract** to create a new authentication policy contract.
2. Edit an existing authentication policy contract by clicking its name.
3. Check the policy contract usage.
4. Optional: On the **Policy Contracts** window, click **Delete** to remove an authentication policy contract, if the authentication policy contract is not in use, or cancel the removal request.

Editing contract information
You can create new policy contracts or edit existing ones from the **Policy Contracts** window.

Steps

1. Go to **Authentication**# **Policies**# **Policy Contracts**.
2. Click **Create New Contract** or select an existing contract.
3. On the **Contract Info** tab, enter or modify the contract name.
4. Click **Next**.

Defining contract attributes
To manage the user attributes in the authentication policy contract, go to **Authentication**# **Policies**# **Policy Contracts**.

About this task

Every authentication policy contract comes with a subject attribute. You can extend the contract with additional attributes as needed.

Steps

▪ If needed, enter an attribute under **Extend the Contract**, and then click **Add**.

> ⓘ **Note:**
>
> Attribute names are case-sensitive and must suit the needs of your partners. Repeat to add more attributes as needed.

▪ Click **Edit**, **Update**, or **Cancel** to modify an existing attribute or undo changes.
▪ Click **Delete** to remove an existing attribute.

Reviewing the policy contract
On the **Summary** tab, you can review the policy contract.

Steps

▪ To keep your changes, click **Save**.
▪ To amend your configuration, click the name of the corresponding tab and then follow the configuration wizard to complete the task.
▪ To discard your changes, click **Cancel**.

Policy fragments

You can manage policy fragments under **Authentication**# **Policies**# **Fragments**. You can create reusable policy fragments and apply them in multiple authentication policies.

Fragments make policies easier to administer, allowing you to extract common patterns that exist among different policies and to manage them in one place. For example, you can create a reusable policy fragment with policy components that you frequently use and apply that fragment in multiple policies. When the authentication requirements need adjustments, you can make those changes in the fragment without updating the policies that reference it.

Fragments can use other fragments; you can chain five or more fragments together. You can also use as many fragments as you want throughout a policy, provided that the way the policy is structured, there aren't over five fragments being processed at one time.

On the **Fragments** window:

▪ Click **Add Fragment** to define a policy fragment.
▪ Click the name of an existing fragment to edit its configuration.
▪ Click **Select Action** to copy or delete an existing fragment.

Defining a policy fragment

You define policy fragments on the **Fragment** window.

Before you begin
Make sure that you have configured at least two policy contracts to function as input and output contracts.

Steps

1. In the **Name** field, type a name for the policy fragment.
2. Optional: Change the identifier for the fragment. This ID will be used to reference input and output attributes in the advanced Expressions fulfillment option. It cannot be changed after the fragment has been created.
3. Optional: Type a description for the fragment.
4. From the **Inputs** list, select the input authentication policy contract that calling members will need to fulfill. The attributes contained in the contract will be available for use throughout the policy.
5. From the **Outputs** list, select the output authentication policy contract that this fragment will fulfill. Calling members will be able to use the values of the attributes contained in the output policy contract.

6. From the **Policy** list, select an IdP adapter, an IdP connection, a selector, or a fragment. (Detailed policy configuration instructions are provided in step 5 in *Defining authentication policies* on page 42.)

> ⓘ **Important**:
>
> As of PingFederate 10.2, you can select **Fragments** as the policy action and then select a policy fragment that you have created. When you select a fragment, click **Fragment Mapping** and use the in-product help links to access the topics that describe how to configure the mapping.

   a. Click **Options**, and select the source and the attribute to be used as the incoming user ID.
   b. Click **Rules**, and define authentication policy rules using attributes from the previous authentication source.
   c. Configure **Fail** and **Success** paths. For a fragment to succeed, you must map it into a LIP or APC based on the output contract. You can also use a fragment in a calling policy and set both of the fragment's exit Fail/Success nodes to Done.

7. When you have completed the fragment's configuration, click **Save**.

Sessions

This topic describes the differences between application and authentication sessions.

Application sessions

Application sessions apply to PingFederate Bridge applications hosted on its user-facing endpoints, such as the profile management page and the grant management endpoints. When the inactivity threshold or the maximum lifetime is reached, PingFederate Bridge redirects previously authenticated users back to the authentication sources, identity provider (IdP) adapter instances or IdP connections, subject to the configuration of authentication sessions.

Authentication sessions

Authentication sessions control when PingFederate Bridge redirects previously authenticated users back to the authentication sources on subsequent requests for browser-based single sign-on (SSO) or PingFederate Bridge applications.

Authentication sessions typically wrap an adapter so that PingFederate Bridge creates the session when user authentication has succeeded. PingFederate Bridge invokes the adapter's authentication logic again only when the session reaches its limits. However, depending on the implementation, an adapter can be aware of an authentication session that wraps it and override this logic. In particular, PingFederate Bridge creates authentication sessions configured for an Identifier First Adapter instance only when the complete single sign-on (SSO) transaction has succeeded. This lets the adapter prompt the user for a different user identifier when a chained adapter authentication fails because, for example, there's a typo in the user identifier.

Session storage options

   When authentication sessions are enabled, PingFederate Bridge maintains session data in memory.

   PingFederate Bridge also supports maintaining session data both in memory and on an external storage. This optional capability allows administrators to support use cases where a longer session duration or a greater resilience against restarts of PingFederate Bridge and browsers is desired. The retrieval and update operations are optimized to provide a fast and seamless user experience. For instance, a retrieval from the external storage is only required when an authentication session is not found in memory.

   > ⓘ **Note:**

Persistent authentication sessions require an external storage. For more information, see *Defining a datastore for persistent authentication sessions*.

Inactivity (idle) timeout and maximum lifetime

When authentication sessions are enabled, an authenticated user is not sent back to the authentication system as long as the user makes another request within the idle timeout window, 60 minutes by default. If the user makes another request within the idle timeout window, the authentication session is extended by the idle timeout value, another 60 minutes by default. For externally stored authentication sessions, this operation is optimized to only send updates to the external storage when the remaining idle timeout window is less than 75%.

An authentication session can be repeatedly extended by multiple requests and remains valid until the maximum timeout value is reached, in which case the user will be redirected back to the authentication system.

ⓘ **Tip:**

The authentication system might or might not challenge the user to complete an authentication process based on its own session management policy or processing logic.

Configuration options

Administrators can enable authentication sessions for all authentication sources, with or without making the authentication sessions persistent, and with or without specifying overrides for selected authentication sources.

Alternatively, administrators can enable authentication sessions for a few selected authentication sources, optionally with their own sets of overrides. The override options include:

- Disable or enable authentication sessions
- Make authentication sessions persistent
- Override the idle timeout, the maximum timeout, or both, in minutes, hours, or days
- Enforce authentication requirement based on authentication context

  Because sessions are tracked with their respective authentication context, administrators can optionally configure PingFederate Bridge to compare the requested authentication context found in the authentication request against the authentication context found in the session. If the values do not match, PingFederate Bridge redirects the user back to the authentication system.

Tracking options for logout

Administrators can optionally configure additional tracking options for logout to control whether PingFederate Bridge should leverage the single logout (SLO) application endpoints to terminate adapter sessions, add sessions to the session revocation list as users sign out, or do both. Publish revoked sessions to provide a secure SLO experience with PingAccess deployments.

Applications

Under **Applications**# **Integration**, you can edit the existing PingOne SP connection, and you can import other connections.

Your PingFederate Bridge license does not allow you to create new SP connections. To do that, you must upgrade to a full PingFederate license.

To upgrade your license, contact Ping Identity Sales via e-mail at *sales@pingidentity.com* or by telephone (+1 877.898.2905 or +1 303.468.2882).

Accessing SP connections

The **SP Connections** window lists all service provider connections and displays up to 20 connections at a time. As needed, you can use the pagination controls to navigate through your connections, narrow connections by protocol type or status, or search for connection by name or ID.

About this task
A connection is included in the search results so long as its name or ID is a partial, case-insensitive match to a search term.

Steps

- Go to **Applications**# **Integration**# **SP Connections**.
- To edit a connection, select the connection by its name. For the setting you want to make a change, select the corresponding window title and then follow the configuration wizard to complete the task.
- To create a connection, click **Create Connection** and follow the on-screen steps.
- To copy a connection, click **Select Action**# **Copy** and then follow the on-screen steps.

  This is most useful if the new connection and the source connection share many common setting values.

  > ⓘ **Note:**
  >
  > PingFederate doesn't include outbound provisioning configurations in connection copies. For more information, see *Configuring outbound provisioning*.

- To export a connection, click **Select Action**# **Export Connection** and then save the XML file as prompted.

  This is useful in situations where you want to make a backup of a connection prior to changing it.
- To import a connection, click **Import Connection**. For more information, see *Importing a connection* on page 100.

  If the connection already exists, you have the option to overwrite the existing connection.

  > ⓘ **Note:**
  >
  > Prior to the import, you can modify the XML file to suit your needs. The XML file can also be imported to another PingFederate Bridge environment acting in the same federation role (IdP) at your site. The source and the target must run the same version of PingFederate Bridge.

- To export metadata for any SAML Browser SSO connection, click **Select Action**# **Export Metadata** and then follow the on-screen instructions.
- To update a SAML Browser SSO connection, click **Select Action**# **Update with Metadata**, then follow the on-screen instructions. For more information, see *Importing SP metadata* on page 66.

  You can update a connection via a metadata XML file or a metadata URL.

  > ⓘ **Important:**
  >
  > The update operation might require additional configuration. Review the connection after the update operation.

- Click the toggle to enable or disable a connection.
- To remove a connection, click **Select Action**# **Delete**.

- To override the verbosity of runtime transaction logging for all SP connections, click **Show Advanced Fields** and the select the desired override option.

Override option	Description
Off	Select this option and let the per-connection **Logging Mode** configuration determine the amount of information PingFederate Bridge records in the runtime transaction log. This is the default selection.
On	Select this option, followed by one of the four logging modes, to set the verbosity of runtime transaction logging for all SP connections. This is most useful when troubleshooting an issue that affects multiple connections.

- To turn off automatic multi-connection error checking, click **Show Advanced Fields**# **Disable Automatic Connection Validation** check box.

  This check box is not selected by default.

  Once selected or cleared, the state of this setting is also reflected on the **Authentication**# **Integration**# **IdP Connections** window.

  For more information about this advanced setting and its impact, see *Configuring automatic connection validation*.
- To keep your changes, click **Save**.
- To discard your changes, click **Cancel**.

Choosing an SP connection type

You can manually create service provider (SP) connections in PingFederate Bridge using browser single sign-on (SSO), WS-Trust security token service (STS), outbound provisioning, or any combination thereof.

About this task

If you are not using a connection template, which pre-configures browser-based SSO, indicate on the **Connection Type** tab whether the connection to this partner is for Browser SSO, WS-Trust STS, outbound provisioning, or any combination of them.

ⓘ **Tip:**

You can add STS, OAuth, and outbound provisioning support to any existing SSO connection, or vice versa, at any time.

ⓘ **Note:**

If your partner's deployment supports multiple protocols and you intend to communicate using more than one, you must set up a separate connection for each protocol. Each connection must use a unique (partner) connection ID.

Steps

1. Go to **Applications**# **Integration**# **SP Connections**.
2. Click **Create Connection**.
3. Select **Do not use a template for this connection**.

4. To configure a connection for secure browser-based SSO, select the **Browser SSO Profiles** check box.

   If you are not using a connection template, you must select the applicable protocol from the list when establishing a new connection.

   For a WS-Federation connection, select the desired token type, either **SAML 1.1**, **SAML 2.0**, or **JWT** (JSON Web Token).

   > ⓘ **Note:**
   >
   > For information about creating a SAML application, see *Configuring a SAML application in PingFederate*.

   > ⓘ **Tip:**
   >
   > If you are creating a WS-Federation connection to Microsoft Windows Azure Pack, select JWT as the token type.

   > ⓘ **Tip:**
   >
   > PingFederate Bridge can encrypt the subject and attributes of SAML 2.0 assertions.
   >
   > For information about configuring encryption policies on a PingFederate Bridge identity provider (IdP), see *Configuring XML encryption policy (SAML 2.0)*.
   >
   > For information about configuring encryption policies on a PingFederate Bridge SP, see *Specifying XML encryption policy (for SAML 2.0)*.

5. Optional: Choose one or both of the following depending on your configuration needs.

Connection Template	Step
WS-TRUST STS	Select the **WS-Trust STS** check box.
Outbound Provisioning	Select **Outbound Provisioning** and then select the provisioning type from the list.

6. If your PingFederate Bridge license manages connections by groups, select a license group for this connection.

   This option is not shown for unrestricted or other types of licenses.

7. To save your settings, click **Next**.

Choosing SP connection options

On the **Connection Options** tab, you can enable browser-based single sign-on (SSO), Attribute Query, or both for the current connection.

Before you begin
For initial steps in creating a service provider (SP) connection, see *Choosing an SP connection type* on page 64.

Steps

1. Choose one or more of the following options.

Connection option	Description
Browser SSO	Select to create a connection for browser-based SSO.
IdP Discovery	Select to enable identity provider (IdP) discovery. This option is only available if you have configured IdP discovery. For more information, see *Configuring standard IdP Discovery* on page 151.
Attribute Query	Select to create a connection that facilitates the SAML 2.0 Attribute Query profile. For more information, see *Attribute Query and XASP* on page 116.

2. To save your changes, click **Next**.

Importing SP metadata

When creating or modifying service provider (SP) connections, PingFederate Bridge allows you to import metadata from an XML file or a metadata URL.

About this task

If you are using one of the SAML protocols without a connection template, you can expedite the setup by one of the following actions:

- Import a metadata file
- Select a metadata URL

When you select a metadata URL, PingFederate Bridge also enables the automatic update option and checks the metadata periodically. If PingFederate Bridge detects changes in the partner's signing certificates for digital signature verification, encryption key, or contact information, it updates the connection automatically. For better housekeeping, the update process removes verification certificates from the connection when the partner no longer maintains them in its metadata. In a clustered environment, PingFederate Bridge automatically replicates verification certificates and encryption key changes to all engine nodes. Offline engine nodes will also consume these changes as they restart and rejoin the cluster. If you prefer to update the connection manually, you can clear the **Enable Automatic Reloading** check box.

You can configure reload frequency at **System**# **Protocol Metadata**# **Metadata Settings**# **Metadata Lifetime** tab. The default reload frequency is daily.

We recommend you turn on notifications for SAML metadata update events at **System**# **Monitoring & Notifications**# **Runtime Notifications**.

ⓘ **Note:**

The notification message provides a list of the applicable items if the metadata contains changes that require additional configuration.

After creating the connection, you can add, remove, or change the metadata URL associated with the connection in the **Metadata URL** tab. In addition, you can toggle the **Enable Automatic Reloading** check box for the connection.

ⓘ **Tip:**

Using a metadata URL with automatic reloading streamlines the configuration process. For example, you can quickly establish a browser SSO connection to an InCommon-participating partner. For more information, see www.incommon.org/participants.

Steps

1. Select from one of the following steps to import or update metadata.

Metadata medium	Steps
Metadata file	**a.** On the **Import Metadata** tab, select the **File** option. **b.** Choose the metadata file, and then click **Next**. ⓘ **Note:** If the metadata contains multiple entries, select the desired partner from the **Select Entity ID** list and click **Next**. ⓘ **Note:** If the metadata file is digitally signed but the verification certificate is provided outside of the metadata, import the metadata verification certificate on the **Import Certificate** tab, and then click **Next**. **c.** On the **Metadata Summary** tab, review the signature information to evaluate the authenticity of the metadata.
Metadata URL	**a.** On the **Import Metadata** tab, select the **URL** option. **b.** Select the metadata from the **Metadata URL** list. ⓘ **Tip:** If the metadata you want is not shown in the list, click **Manage Partner Metadata URLs**. For more information, see *Manage Partner metadata URLs* on page 109. **c.** Optionally, clear the **Enable Automatic Reloading** check box to disable automatic update. ⓘ **Note:** A warning will display if you do not have runtime notifications enabled. To enable these notifications, go to **System**# **Monitoring & Notifications**# **Runtime Notifications** and

Metadata medium	Steps
	select the **Notification for SAML Metadata Update Events** box.
	d. Click **Load Metadata**.
	ⓘ **Note:**
	If the metadata contains multiple entries, select the desired partner from the **Select Entity ID** list and click **Next**.
	ⓘ **Note:**
	If there is a digital signature error, click **Manage Partner Metadata URLs** to resolve the issue.

2. Click **Next**.

Identifying the SP

On the **General Info** tab, you provide your partner's unique federation identifier, the display name of the connection, and some other optional information, such as virtual server IDs, contact information, and logging mode for runtime transaction logging.

Steps

1. For information on initial steps for managing SP connections, see *Choosing an SP connection type* on page 64.

2. Provide the basic information to identify your partner.

See the following table for more information.

Field	Description
Partner's Entity ID, Issuer, Partner's Realm, or Connection ID (Required)	The published, protocol-dependent, unique identifier of your partner.
	For a SAML 2.0 connection, this is your partner's SAML Entity ID.
	For a SAML 1.x connection, this is the `Audience` your partner advertises. This ID may have been obtained out-of-band or using a SAML metadata file.
	For a WS-Federation connection, this is your partner's Realm.
	For a security token service (STS)-only connection, you can designate any unique identifier.
Connection Name (Required)	A plain-language identifier for the connection. For example, a company or department name. This name is displayed in the connection list on the administrative console.
Virtual Server IDs	If you want to identify your server to this connection partner using an ID other than the one you specified at **System**# **Server**# **Protocol Settings**# **Federation Info**, enter a virtual server ID in this field and click **Add**.
	Enter additional virtual server IDs as needed.

Field	Description
Base URL	The fully qualified host name and port on which your partner's federation deployment runs. For example, *https://www.example.com:9031*. This entry is an optional convenience, allowing you to enter relative paths to specific endpoints, instead of full URLs, during the configuration process.
Company	The name of the partner company to which you are connecting.
Contact Name	The contact person at the partner company.
Contact Number	The phone number of the contact person at the partner company.
Contact Email	The email address for the contact person at the partner company.
Application Name	The name of the application, accessible through the IdP Adapter interface IdpAuthenticationAdapterV2 in the PingFederate Java SDK.
	This field is not applicable to an STS-only connection.
Application Icon URL	The URL of the application icon, accessible through the IdP Adapter interface IdpAuthenticationAdapterV2 in the PingFederate Java SDK.
	Note that this field is not applicable to an STS-only connection.
Logging Mode	The level of transaction logging applicable for this connection.

3. After entering the relevant identification information, click **Next**.

Configure IdP Browser SSO

Browser-based single sign-on (SSO), also known as Browser SSO, relies on a user's web browser and HTTP requests to broker identity-federation messaging in XML or JSON web token (JWT) between an identity provider (IdP) and a service provider (SP).

Go to **Applications**# **Integration**# **SP Connections** to access an existing or create a new SP connection. For more information, see *Accessing SP connections* on page 63.

From the **Browser SSO** tab inside your SP connection instance, click **Configure Browser SSO** and follow the steps below based on your federation protocol.

> ⓘ **Tip:**
>
> Many steps involved in setting up a federation connection are protocol-independent. They are required steps for all connections, regardless of the associated standards. For more information, see *Federation roles*.
>
> Some steps are required under the applicable protocol, while others are optional. Still others are required only based on certain selections. The administrative console determines the required and optional steps based on the protocol and dynamically presents additional requirements or options based on selections.
>
> The following sections provide sequential information about every step you might encounter while configuring browser-based SSO, depending on the protocol you are using for a particular connection.

SAML 2.0 configuration steps

- *Choosing SAML 2.0 profiles* on page 70
- *Setting an SSO token lifetime* on page 71
- *Configuring SSO token creation* on page 72

  - *Choosing an identity mapping method for IdP SSO* on page 72
  - *Setting up an attribute contract* on page 74
  - *Managing authentication source mappings* on page 78

SAML 1.x configuration steps

WS-Federation configuration steps

After configuring SSO settings, you will normally need to configure authentication credentials, the range of which depends on your SSO selection. For more information, see *Configuring credentials* on page 98.
You might need to complete further configuration tasks for new or modified connections, depending on the selected options on the **Connection Options** tab.

Choosing SAML 2.0 profiles
A SAML profile is the message-interchange scenario that you and your federation partner have agreed to use. SAML binding, by contrast, is the transport protocol of SAML messages.

About this task

On the **SAML Profiles** tab, select one or more SAML 2.0 profiles for your IdP Browser SSO configuration.

> ⓘ **Note:**
>
> The **SAML Profiles** tab is not shown for SAML 1.x connections because identity provider (IdP) single sign-on (SSO) is assumed, single logout (SLO) profiles are not supported, and the server supports the "destination-first" (SP-initiated) profile SSO automatically. This window is also not presented for WS-Federation connections because profile selection is not required.

> ⓘ **Note:**
>
> When configuring a local loopback connection, in which one PingFederate instance is both the identity provider and the service provider, disable the IdP-Initiated SLO and SP-Initiated SLO options on the Browser SSO window's SAML Profiles tab. These options determine whether SAML logout requests should be sent to the partner during the SLO flow. Those requests aren't necessary and can cause unexpected behavior when the partner connection exists locally. All local sessions for loopback connections are terminated during the SLO flow without the need to send SAML requests.

For SAML 2.0, PingFederate Bridge supports all IdP- and SP-initiated SSO and SLO profiles. For more information on typical SSO and SLO profile configurations, including illustrations, see *SAML 2.0 profiles*.

Steps

1. Go to **Applications**# **Integration**# **SP connections**.
2. Click on the SP connection you want to configure. For more information, see *Accessing SP connections* on page 63.
3. On the **Browser SSO** tab, click **Configure Browser SSO**.
4. Select either **IdP-Initiatied SSO** or **SP-Initiated SSO** or both, depending on your partner agreement.

   You must select at least one SSO profile.
5. Select either **IdP-Initiated SLO** or **SP-Initiated SLO** or both, depending on your partner agreement.

   SLO profile options are only enabled after you choose an SSO profile.

IdP Connections | IdP Connection | Browser SSO

| SAML Profiles | User-Session Creation | Protocol Settings | Summary |

A SAML Profile defines what kind of messages may be exchanged between an Identity Provider (IdP) and a Service Provider (SP), and how the messages are transported (bindings). As an SP, you configure this information for your IdP connection.

Single Sign-On (SSO) Profiles	Single Logout (SLO) Profiles
✔ IDP-INITIATED SSO	✔ IDP-INITIATED SLO
☐ SP-INITIATED SSO	☐ SP-INITIATED SLO

6. Click **Next** to save your changes.

Setting an SSO token lifetime
Identity-federation standards require a window of time during which a SSO token is considered valid. Each SSO token has an issuance time-stamp element and elements indicating the allowable lifetime of the SSO token before and after the issuance time stamp.

Before you begin
For previous steps in configuring Browser SSO, see *Choosing SAML 2.0 profiles* on page 70. For more information about managing service provider (SP) connections, see *Accessing SP connections* on page 63.

About this task

PingFederate Bridge gives you the option to change the valid lifetime of the single sign-on (SSO) token.

Steps

1. Optional: Override the default values for the following fields.

Field	Description
Minutes Before	The amount of time before the SSO token was issued during which it is to be considered valid.
Minutes After	The amount of time after the SSO token was issued during which it is to be considered valid.

The default value is 5 minutes for both fields.

2. Click **Next** to save your changes.

Configuring SSO token creation

As an identity provider (IdP), you must specify how PingFederate Bridge obtains user-authentication information and use it to create single sign-on (SSO) tokens appropriate for your service provider (SP) partner, including additional user attributes as needed.

About this task

If you are a federation hub bridging a service provider to one or more identity providers, you can associate one or more authentication policy contracts to the SP connection. For more information, see *Federation hub use cases*.

The configuration involves choosing an identity-mapping method, if applicable; establishing an attribute contract, as needed; and mapping one or more IdP adapter instances, authentication policy contracts, or both.

Steps

1. Go to **Applications**# **Integration**# **SP Connections**.

2. Click on the SP connection that you want to configure.

3. Follow the steps to reach the **Browser SSO** tab for your connection. For more information, see *Configure IdP Browser SSO* on page 69.

4. On the **Assertion Creation** tab, click **Configure Assertion Creation**.

Choosing an identity mapping method for IdP SSO

In the **Identity Mapping** window, you choose the type of name identifier your partner requires. Your selection might affect the way that the service provider (SP) looks up and associates your users at the SP site. You and the SP should decide in advance which option to use.

The choices of name-identifier types depend on whether you use the SAML or WS-Federation protocol. For more information, see one of the following.

- *Selecting a SAML Name ID type* on page 73
- *Selecting a WS-Federation Name ID type* on page 74

> ⓘ **Note:**
>
> The **Identity Mapping** window does not apply for connections using the WS-Federation protocol in conjunction with JSON web token (JWT)-based single sign-on (SSO) tokens. Instead, work with the SP to define an attribute contract that it can use to map users to accounts at the SP site.

Selecting a SAML Name ID type
You can choose a name identifier for your SAML Browser single sign-on (SSO) configuration on the
identity Mapping tab. The type of name identifier you select affects how your service provider (SP)
partner makes use of account mapping or account linking.

Before you begin
For previous steps in configuring Browser SSO, see *Configure IdP Browser SSO* on page 69. For more
information about managing service provider (SP) connections, see *Accessing SP connections* on page
63.

About this task

If your SP uses account linking, establishing an attribute contract is not required. However, depending
on your agreement, you can choose to supplement the account link with an attribute contract. In this
configuration, the account link is used to determine the user's identity, while the additional attributes
might be used for authorization decisions, customized web pages, and so on, at the SP site. For more
information, see *User attributes*.

> ⓘ **Important**:
>
> If you change your configuration to use account linking without additional attributes, any existing attribute
> contract will be discarded in favor of the new configuration.

Steps

1. Select the type of name identifier that you and your SP have agreed to use.

Option	Description
Standard	Select if you want to send a known attribute to identify a user, for example, a username or an email address.
	In this scenario, the SP often uses account mapping to identify the user locally.
Pseudonym	Select if you and the SP have agreed to use a unique, opaque persistent name identifier, which cannot be traced back to the user's identity at the IdP.
	The SP might also use the identifier for account linking to make a persistent association between the user and a specific local account.
	Select the **Include attributes in addition to the pseudonym** box if you want to set up an attribute contract to use in conjunction with an opaque identifier. For more information, see *Setting up an attribute contract* on page 74.
Transient	Select **Transient** to enhance the privacy of a user's identity. Unlike a pseudonym, a transient identifier is different each time a user initiates SSO.
	An example application for this selection might be when an SP provides generalized group accounts based on organizational rather than individual identity.
	Select the **Include attributes in addition to the transient identifier** box if you want to set up an attribute contract to use in conjunction with an opaque identifier. For more information, see *Setting up an attribute contract* on page 74.

2. Click **Next** to save your changes.

Next steps
If you opted to include attributes in your name identifier, your next step will be to define the attributes. For more information, see *Setting up an attribute contract* on page 74. Otherwise proceed to *Managing authentication source mappings* on page 78.
Selecting a WS-Federation Name ID type
You can choose a name identifier for your WS-Federation Browser single sign-on (SSO) configuration on the **Identity Mapping** tab. Your selection might affect the way the service provider (SP) looks up and associates your users to their local accounts.

Before you begin
For previous steps in configuring Browser SSO, see *Configure IdP Browser SSO* on page 69. For more information about managing service provider (SP) connections, see *Accessing SP connections* on page 63.

About this task
The **Identity Mapping** window is not applicable to connections using the WS-Federation protocol in conjunction with JSON web token (JWT)-based SSO tokens. Instead, work with the SP to define an attribute contract that it can use to map users to accounts at the SP site.

Steps

1. Select the type of name identifier that you and your SP have agreed to use.

Option	Description
Email Address	This attribute is commonly used as a unique identifier for SSO and single logout (SLO). Make this selection, for example, if a user logs in using an email address or if the information is available for lookup in a local datastore.
User Principal Name	The username or other unique ID of the subject initiating the transaction. Make this selection, for example, if a username will be available from the current user session as part of a cookie or can be derived from a local datastore.
Common Name	This selection provides for anonymous SSO to your SP, generally using a hard-coded generalized sign on. Make this selection if your partner agreement involves a many-to-one use case, such as if the SP has a group account set up for all users in a particular domain.

2. Click **Next** to save your changes.

Setting up an attribute contract
An attribute contract is the set of user attributes that you and your partner have agreed will be sent in the single sign-on (SSO) tokens for this connection.

About this task

You specify the attributes for the name identifier on your WS-Federation or, optionally, for your SAML configuration on the **Attribute Contract** tab. For more information, see *Attribute contracts*.

WS-Federation connections require you to define attribute contracts. For SAML connections, attribute contracts are optional if you are sending either pseudonym or transient identifiers to the partners. For more information, see *Selecting a SAML Name ID type* on page 73.

When establishing an attribute contract, you can change the name format when certain conditions are met. The following table summarizes the conditions and the possible actions that you can perform on the **Attribute Contract** tab.

Protocol	Identity mapping	Attribute contract	SAML_SUBJECT	Additional attributes
SAML 2.0 or SAML 1.1	Standard	Required	Built-in. Subject name format can be changed by selecting a value from a list.	Optional. Attribute name format can be changed by selecting a value from a list.
SAML 2.0 or SAML 1.1	Pseudonym or Transient	Required only if the **Include attributes ...** check box is selected on the **Identity Mapping** window. Otherwise the **Attribute Contract** window is not shown.	Assumed and cannot be added as an additional attribute.	At least one is required. Attribute name format can be changed by selecting a value from a list.
SAML 1.0	Standard	Required	Built-in. Subject name format can be changed by selecting a value from a list.	Optional. There is no attribute name format.
SAML 1.0	Pseudonym or Transient	Required only if the **Include attributes ...** check box is selected on the **Identity Mapping** window. Otherwise the **Attribute Contract** window is not shown.	Assumed and cannot be added as an additional attribute.	At least one is required. There is no attribute name format.
WS-Federation in conjunction with SAML 1.1 as the token type	Email address, user principal name, or common name	Required	Built-in. There is no subject name format.	Optional. Attribute name format can be changed by selecting a value from a list.
WS-Federation in conjunction with SAML 2.0 as the token type	Email address, user principal name, or common name	Required	Built-in. There is no subject name format.	Optional. Attribute name format can be changed by selecting a value from the list.

Protocol	Identity mapping	Attribute contract	SAML_SUBJECT	Additional attributes
WS-Federation in conjunction with JWT as the token type	Not applicable	Required	Not applicable	At least one is required. There is no attribute name format.

ⓘ **Tip:**

If you are creating or updating a SAML service provider (SP) connection, consider using the partner's metadata to do so. If the metadata contains the required information, PingFederate Bridge automatically populates the attribute contract for you. For more information, see *Importing SP metadata* on page 66.

Steps

1. Follow the required steps to create an SSO token depending on your federation protocol. For more information, see *Configure IdP Browser SSO* on page 69.
2. If you are using a SAML protocol, on the **Identity Mapping** tab you must select either **Pseudonym** or **Transient**, and also select the **Include Attributes** box to access the **Attribute Contract** tab. For more information, see *Selecting a SAML Name ID type* on page 73.
3. Optional: Click the **Attribute Name Format** drop-down to select a different format for the built-in subject identifier, **SAML_SUBJECT**.

   Applicable if you and the SP have agreed to a specific format. For more information, see *Attribute contracts*.

ⓘ **Note:**

As needed, you can customize name-format alternatives in the `<pf_install>`/pingfederate/`server/default/data/config-store/custom-name-formats.xml` configuration file. Restart PingFederate Bridge to activate any changes made to this file.

---

4. Extend the contract with additional attributes.
   a. Enter the name of an additional attribute in the text field under **Extend the Contract**.

      Attribute names are case-sensitive and must correspond to the attribute names expected by your partner.

ⓘ **Tip:**

You can add a special attribute, SAML_AUTHN_CTX, to indicate to the SP, if required, the type of credentials used to authenticate to the identity provider (IdP) application.

The value of this attribute can then be mapped later on the **Attribute Contract Fulfillment** window. For more information, see *Configuring contract fulfillment for IdP Browser SSO* on page 82. The mapped value overrides the authentication context provided by the IdP adapter instance or the Requested AuthN Context Authentication Selector instance, through an authentication policy. If no authentication context is provided by the SAML_AUTHN_CTX attribute, the IdP adapter instance,

---

or the Requested AuthN Context Authentication Selector instance, PingFederate Bridge sets the authentication context as follows:

- For SAML 1.x `urn:oasis:names:tc:SAML:1.0:am:unspecified`
- For SAML 2.0 `urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified`

ⓘ **Tip:**

If you are configuring a WS-Federation connection to Microsoft Windows Azure Pack, add upn to the JWT's attribute contract.

ⓘ **Tip:**

If you are configuring a SAML connection to an InCommon participant (see *Incommon federation participants*), the attribute contract might contain or require attributes such as `urn:oid:0.9.2342.19200300.100.1.3` and `urn:oid:2.5.4.42`, which are standard names under various specifications, such as *RFC4524* and *RFC4519* . The following table describes a subset of the object IDs referenced by the most common attributes used by InCommon participants.

Object ID value	Description
0.9.2342.19200300.100.1.3	mail
1.3.6.1.4.1.5923.1.1.1.6	eduPersonPrincipalName
1.3.6.1.4.1.5923.1.1.1.7	eduPersonEntitlement
1.3.6.1.4.1.5923.1.1.1.9	eduPersonScopedAffiliation
1.3.6.1.4.1.5923.1.1.1.10	eduPersonTargetedID
2.5.4.3	cn
2.5.4.4	sn
2.5.4.10	o
2.5.4.42	givenName
2.16.840.1.113730.3.1.241	displayName

For other attributes, see the metadata from your partner. The FriendlyName values, if available, should provide additional information about the attributes. Alternatively, third-party resources, such as *https://www.ldap.com/ldap-oid-reference* and *http://www.oid-info.com/*, might help as well.

b. Select an attribute name format from the list.

Applicable if you and the SP have agreed to a specific format. For more information, see *Attribute contracts*.

ⓘ **Note:**

As needed, you can customize name-format alternatives in the *`<pf_install>`*`/pingfederate/server/default/data/config-store/custom-name-formats.xml` configuration file. Restart PingFederate Bridge to activate any changes made to this file.

---

c. Click **Add**.
d. Repeat until all desired attributes are defined.

5. Optional: Click **Edit** to change the configuration of an existing attribute.
6. Optional: Click **Delete** to remove an existing attribute.

7. Click **Next** to save changes.

Managing authentication source mappings
On the **Authentication Source Mapping** tab, you can map identity provider (IdP) adapters and authentication policies to authenticate users to your service provider (SP).

About this task

IdP adapters are responsible for handling user authentication as part of an single sign-on (SSO) operation. A configured adapter in PingFederate Bridge is known as an adapter instance.

In a basic scenario, you map an IdP adapter instance to a SP connection on the **Authentication Source Mapping** tab and complete its mapping configuration through a series of sub tasks. When a user starts an SSO request, the corresponding IdP adapter is triggered to authenticate the user. Upon successful authentication, PingFederate Bridge creates and sends an SSO token to the SP based on the connection settings. As needed, you can map multiple IdP adapter instances to an SP connection, the same IdP adapter instance to multiple SP connections, or a combination of them.

If you use authentication policies to route users through a series of authentication sources and end each successful policy path with an authentication policy contract (APC), you can map the APC to your connection. Like IdP adapter instances, you can map multiple APCs to an SP connection, the same APC to multiple SP connections, or a combination of them.

> ⓘ **Tip:**
>
> For more information about authentication policies and contracts, see *Authentication policies*.

You can also map one or more APCs to an SP connection to bridge a service provider to one or more identity providers. In this scenario, PingFederate Bridge is a federation hub for both sides. PingFederate Bridge uses APCs to associate this SP connection with the applicable IdP connections to the identity providers. Each APC has its own set of attributes which you map values to the SSO tokens.

> ⓘ **Tip:**
>
> For more information about the federation hub, see *Federation hub use cases*.

Regardless of how many IdP adapter instances and APCs are mapped to an SP connection, PingFederate Bridge uses only one adapter instance or policy path to authenticate a user. You can leave the decision to the users or create authentication policies to mandate authentication requirements. Because each adapter instance or APC could return different user attributes, each mapping must define how the attribute contract is fulfilled in its mapping configuration.

Steps

1. For initial steps to configure SP connections, see *Accessing SP connections* on page 63.
2. For initial steps to configure Browser SSO, see *Configure IdP Browser SSO* on page 69.
3. For initial steps to configure assertion creation. see *Configuring SSO token creation* on page 72.

4. On the **Authentication Source Mapping** tab, select one of the following.

   - Click **Map New Adapter Instance** to map a new IdP Adapter instance. For more information, see *Mapping an adapter instance* on page 79.
   - Click **Map New Authentication Policy** to map a new APC. For more information, see *Mapping an authentication policy*
   - Click on an existing instance to edit its configuration.
   - Click **Delete** to remove an existing adapter instance or APC. Click **undelete** to cancel the removal request

   When authentication sources, such as IdP adapter instances or connection mapping contracts, are restricted to certain virtual server IDs, the allowed IDs are displayed under **Virtual Server IDs**.

5. When your authentication sources have been mapped, click **Next** save your changes.

Mapping an adapter instance
After extending your attribute contract, you can map adapter instances on the **Authentication Source Mapping** tab.

Steps

1. For initial steps, see *Managing authentication source mappings* on page 78.
2. On the **Authentication Source Mapping** tab, click **Map New Adapter Instance**.
3. On the **Adapter Instance** tab, select an adapter instance from the **Adapter Instance** drop-down list.
4. Optional: If you want to customize one or more adapter settings for this connection alone, select the **Override Instance Settings** check box.

   > ⓘ **Note:** If you are editing a currently mapped adapter instance, you can toggle **Override Instance Settings**. Clearing it removes all previously overridden settings for this connection. Selecting it provides you the opportunity to customize adapter settings specifically for this connection.

   > ⓘ **Tip:**
   >
   > Alternatively, you can create child adapter instances of a base adapter instance (with overrides) so that customized settings can be applied to several connections. For more information, see *Hierarchical plugin configurations*.

   Selecting the **Override instance settings** box will add the **Override Instance** tab to the navigation bar. For more information, see *Overriding an IdP adapter instance* on page 79

5. Click **Next**, and refer to the following topics to complete the configuration.

Overriding an IdP adapter instance
On the **Override Instance** window, you can start a series of sub tasks to override adapter settings specifically for this connection.

About this task

> ⓘ **Note:**
>
> Any changes to the base adapter instance are propagated to a connection as long as you don't override those changes.

Steps

1. For initial steps to configure authentication source mapping, see *Managing authentication source mappings* on page 78.

2. On the **Adapter Instance** tab, click **Override Instance Settings**.

3. On each of the settings tabs, select the **Override** check box, make your changes, and then click **Next**.

> ⓘ **Note:**
>
> If you are editing a currently mapped adapter instance, click **Override Instance Settings** to reconfigure any overridden settings for this connection. You can also remove all overridden settings on a per-window basis by clearing the **Override** check box near the top of the window.

The override setting windows are functionally identical to those used for creating a new adapter instance. For more information, see *Managing IdP adapters* on page 23.

4. When you are finished, click **Done** to proceed to *Selecting an attribute mapping method* on page 80.

Restricting an authentication source to certain virtual server IDs
On the **Virtual Server IDs** tab, when you multiplex one connection for multiple environments, you can enforce authentication requirements by restricting an authentication source to certain virtual server IDs.

About this task

Authentication sources are unrestricted by default. For more information, see *Multiple virtual server IDs*

Steps

1. Select the **Restrict Virtual Server IDs** check box.

2. Select one or more virtual server IDs that you want to allow for this authentication source.

Results

If you are editing a currently mapped adapter instance or authentication policy contract (APC), you can toggle the **Restrict Virtual Server IDs** setting. You can also change the allowed virtual server IDs.

Selecting an attribute mapping method
On the **Mapping Method** tab, you can select if and how PingFederate Bridge should query local datastores to help fulfill the attribute contract in conjunction with attribute values from the authentication source.

About this task

To determine whether you need to look up additional values, compare the attribute contract against the adapter contract or the authentication policy contract. If the attribute contract requires more information, you must determine whether local datastores can supply it.

> ⓘ **Tip:**
>
> Alternatively, you can configure datastore queries as part of the fulfillment configuration for the applicable identity provider (IdP) adapter contract or authentication policy contract. If so, you do not need to set up datastore query on the connection level.
>
> For more information, see *Defining the IdP adapter contract* on page 26 or *Applying policy contracts or identity profiles to authentication policies*.

Steps

1. For initial steps to configure IdP adapter instances, see *Mapping an adapter instance* on page 79.

2. On the **Mapping Method** tab, select one of the following options.

Mapping method	Description
Retrieve additional attributes from multiple data stores using one mapping	Select to configure one or more datastores to look up attributes for a single mapping.
Retrieve additional attributes from a data store	Select to define alternate datastores to look up attributes and a failsafe mapping configuration. ⓘ **Note:** When this option is selected, the token authorization framework, through issuance criteria, does not apply. For more information, see *Token authorization* and *Selecting an attribute mapping method*.
Use only the adapter contract values in the SAML assertion	Select if you do not require connection-level datastore query.

3. Click **Next** to save changes and proceed to the next tab.

If you opted to require datastore queries, see *Configuring attribute sources and user lookup* on page 81. If not, see *Configuring contract fulfillment for IdP Browser SSO* on page 82.

Configuring attribute sources and user lookup
Attribute sources are specific datastore or directory locations containing information that might be needed for the attribute contract. You can use more than one attribute source when mapping values to the attribute contract.

About this task

The order in which attribute sources are listed affects the queries differently based on the selection made on the **Mapping Method** tab. For more information, see *Selecting an attribute mapping method* on page 80.

Retrieve additional attributes from multiple data stores using one mapping

If you plan on using the result of a query as an input to a subsequent query, stack your attribute sources accordingly.

Retrieve additional attributes from a data store

As soon as a query succeeds, PingFederate Bridge moves on to the next task, contract fulfillment. Therefore you should prioritize the attribute sources.

Steps

1. Click **Add Attribute Source** and then follow a series of sub tasks to complete the configuration.

2. See *Choosing a datastore* on page 28 for instructions on configuring and adding attribute sources.

3. Repeat as necessary to add additional sources.

Results

If you are editing a currently mapped adapter instance or authentication policy contract, you can add, remove, or reorder attribute sources, which might require additional configuration changes in subsequent tasks.

Configuring contract fulfillment for IdP Browser SSO

On the **Attribute Contract Fulfillment** tab, you can map values to the attributes defined for the contract. These are the values that will be included in the single sign-on (SSO) tokens sent to the service provider (SP).

Before you begin

For initial steps to configure identity provider (IdP) adapter instances or authentication policy contracts (APC), see *Managing authentication source mappings* on page 78.

About this task

If you are bridging one or more identity providers to a service provider, map values to an authentication policy contract. For more information, see *Federation hub use cases*.

At runtime, an SSO operation fails if PingFederate Bridge cannot fulfill the required attribute.

On the **Attribute Contract Fulfillment** tab, you must complete the following steps for each attribute contract.

Steps

1. Select a **Source** from the drop-down.
2. Select a **Value** from the drop-down or enter a **Value** in the text field. See the following for more information.

   ▪ **Adapter** or **Authentication Policy Contract** (the authentication source)

   When selected, the **Value** list is populated with attributes from the authentication source. Select the desired attribute from the list. At runtime, the attribute value from the authentication source is mapped to the value of the attribute in the SSO token.

   For example, to map the value of the HTML Form Adapter's username attribute as the value of the SAML_SUBJECT attribute on the contract, select **Adapter** from the **Source** list and **username** from the **Value** list.

   ▪ **Context**

   When selected, the **Value** list populates with the available context of the transaction. Select the desired context from the list. At runtime, the context value is mapped to the value of the attribute in the SSO token.

   > ⓘ **Important:**
   >
   > If you are configuring an SP connection to bridge one or more identity providers to a service provider, consider mapping the original issuer of the assertions into an attribute by selecting **Context** as the source and **Authenticating Authority** as the value. This is important when bridging multiple

identity providers to one service provider, where the service provider should take the information about the original issuer into consideration before granting access to protected resources.

For more information, see *Bridging multiple IdPs to an SP*.

ⓘ **Note:**

Because the **HTTP Request** context value is retrieved as a Java object rather than text, use OGNL expressions to evaluate and return values (see **Expression**).

- **LDAP**, **JDBC**, or **Other**

  When selected, the **Value** list populates with attributes that you have selected in the attribute source configuration. Select the desired attribute from the list. At runtime, the attribute value from the attribute source is mapped to the value of the attribute in the SSO token.

- **Expression**

  When enabled, this option provides more complex mapping capabilities, such as transforming incoming values into different formats. Select **Expression** from the **Source** list, click **Edit** under **Actions**, and compose your OGNL expressions. All variables available for text entries are also available for expressions. For more information, see **Text**.

  Expressions are not enabled by default. For more information about enabling and editing OGNL expressions, see *Attribute mapping expressions* on page 37.

- **No Mapping**

  Select this option to ignore the **Value** field.

- **Text**

  When selected, the text you enter is mapped to the value of the attribute in the single sign-on tokens at runtime. You can mix text with references to any of the values from the authentication source using the ${*attribute*} syntax.

  You can also enter values from your datastore, when applicable, using this syntax:

  ```
  ${ds.attr-source-id.attribute}
  ```

  where *attr-source-id* is the attribute source ID value and *attribute* is one of the selected attributes in the attribute source configuration.

  Note that when using alternate data stores with a failsafe mapping, the attribute source ID value is not applicable. Use the following syntax instead:

  ```
  ${ds.attribute}
  ```

ⓘ **Tip:**

Two other text variables are also available: ${SAML_SUBJECT} and ${TargetResource}. SAML_SUBJECT is the initiating user (or other entity). TargetResource is a reference to the protected application or other resource for which the user requested SSO access; the ${TargetResource} text variable is available only if specified as a query parameter for the relevant endpoint (either as TargetResource for SAML 2.0 or TARGET for SAML 1.*x*).

  There are also a variety of reasons why you might hard code a text value. For example, if the SP web application provides a service based on the name of your organization, you might provide that attribute value as a constant.

3. After all attributes have been mapped, click **Next** to save changes.

> ⓘ **Note:**
>
> If you are editing a currently mapped adapter instance or APC, you can update the mapping configuration, which might require additional configuration changes in subsequent tasks.

Configuring default contract fulfillment for IdP Browser SSO
On the **Attribute Contract Fulfillment** tab, you can define the default attributes PingFederate Bridge will send to the service provider (SP) in case of failure to complete the attribute contract.

Before you begin

For initial steps to configure identity provider (IdP) adapter instances or authentication policy contracts (APC), see *Managing authentication source mappings* on page 78.

If you have selected the failsafe option on the *Mapping Method* tab and the **Send user to SP using default list of attributes** option on the *Failsafe Attribute Source* tab, define the default values that should be sent in the single sign-on (SSO) tokens to the SP.

About this task
On the **Attribute Contract Fulfillment** tab, you must complete the following steps for each adapter instance or APC.

Steps

1. Select a source from the **Source** drop-down list.

2. Select a source from the **Source** list and then choose or enter a value. You must map all attributes. See the following table for more information.

- **Adapter** or **Authentication Policy Contract** (the authentication source)

   When selected, the **Value** list is populated with attributes from the authentication source. Select the desired attribute from the list. At runtime, the attribute value from the authentication source is mapped to the value of the attribute in the SSO token.

   For example, to map the value of the HTML Form Adapter's username attribute as the value of the SAML_SUBJECT attribute on the contract, select **Adapter** from the **Source** list and **username** from the **Value** list.

- **Context**

   When selected, the **Value** list populates with the available context of the transaction. Select the desired context from the list. At runtime, the context value is mapped to the value of the attribute in the SSO token.

   > ⓘ **Important:**
   >
   > If you are configuring an SP connection to bridge one or more identity providers to a service provider, consider mapping the original issuer of the assertions into an attribute by selecting **Context** as the source and **Authenticating Authority** as the value. This is important when bridging multiple

identity providers to one service provider, where the service provider should take the information about the original issuer into consideration before granting access to protected resources.

For more information, see *Bridging multiple IdPs to an SP*.

> ⓘ **Note:**
>
> Because the **HTTP Request** context value is retrieved as a Java object rather than text, use OGNL expressions to evaluate and return values (see **Expression**).

- **Expression**

  When enabled, this option provides more complex mapping capabilities, such as transforming incoming values into different formats. Select **Expression** from the **Source** list, click **Edit** under **Actions**, and compose your OGNL expressions. All variables available for text entries are also available for expressions. For more information, see **Text**.

  Expressions are not enabled by default. For more information about enabling and editing OGNL expressions, see *Attribute mapping expressions* on page 37.
- **No Mapping**

  Select this option to ignore the **Value** field, causing no value selection to be necessary.
- **Text**

  When selected, the text you enter is mapped to the value of the attribute in the single sign-on tokens at runtime. You can mix text with references to any of the values from the authentication source using the ${*attribute*} syntax.

  > ⓘ **Tip:**
  >
  > Two other text variables are also available: ${SAML_SUBJECT} and ${TargetResource}. SAML_SUBJECT is the initiating user (or other entity). TargetResource is a reference to the protected application or other resource for which the user requested SSO access; the ${TargetResource} text variable is available only if specified as a query parameter for the relevant endpoint (either as TargetResource for SAML 2.0 or TARGET for SAML 1.*x*).

3. After all attributes have been mapped, click **Next** to save changes.

Defining issuance criteria for IdP Browser SSO
Configure the criteria that PingFederate Bridge uses to determine user authorization to access service provider (SP) resources.

About this task

On the **Issuance Criteria** tab, define the criteria that must be satisfied in order for PingFederate Bridge to process a request further. This token authorization feature provides the capability to conditionally approve or reject requests based on individual attributes.

> ⓘ **Note:**
>
> The **Issuance Criteria** tab does not appear if you have chosen the failsafe option on the **Mapping Method** tab. For more information, see *Selecting an attribute mapping method* on page 80.

Begin this optional configuration by adding a criterion. Choose the source that contains the attribute to be verified. Some sources, such as **Mapped Attributes**, are common to almost all use cases. Other sources, such as **JDBC**, depend on the type of configuration. Irrelevant sources are automatically hidden. After you select a source, choose the attribute to be verified. Depending on the selected source, the available attributes or properties vary. Finally, specify the comparison method and the desired, compared-to, value.

If you define multiple criteria, all criteria must be satisfied for PingFederate Bridge to move a request to the next phase. A criterion is satisfied when the runtime value of the selected attribute matches or does not match the specified value depending on the chosen comparison method. The multi-value contains and multi-value does not contain comparison methods are intended for attributes that might contain multiple values. Such criterion is considered satisfied if one of the multiple values matches or does not match the specified value. Values are compared verbatim. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions.

> ⓘ **Important:**
>
> When you multiplex one connection for multiple environments, consider using attribute mapping expressions to verify the virtual server ID in conjunction with other conditions, such as group membership information, to protect against unauthorized access. For more information, see *Multiple virtual server IDs* and *Issuance criteria and multiple virtual server IDs*.

> ⓘ **Note:**
>
> All criteria defined must be satisfied or evaluated as true for a request to move forward. As soon as one criterion fails, PingFederate Bridge rejects the request and returns an error message.

Steps

1. From the **Source** list, select the attribute's source.

Source	Description
Adapter or Authentication Policy Contract	Select to evaluate attributes from an identity provider (IdP) adapter instance or an authentication policy contract.
Context	Select to evaluate properties returned from the context of the transaction at runtime. > ⓘ **Note:** > The **HTTP Request** context value is retrieved as a Java object rather than text. For this reason, attribute mapping expressions are more appropriate to evaluate and return values.
JDBC, LDAP, or other types of datastore (if configured)	Select to evaluate attributes returned from a data source.
Mapped Attributes	Select to evaluate the mapped attributes.

2. From the **Attribute Name** list, select the attribute to be evaluated.

3. From the **Condition** list, select the comparison method.

Available methods:

- **equal to**
- **equal to (case insensitive)**
- **equal to DN**
- **not equal to**
- **not equal to (case insensitive)**
- **not equal to DN**
- **multi-value contains**
- **multi-value contains (case insensitive)**
- **multi-value contains DN**
- **multi-value does not contain**
- **multi-value does not contain (case insensitive)**
- **multi-value does not contain DN**

> ⓘ **Note:**
>
> The first six conditions are intended for single-value attributes. Use one of the **multi-value ...** conditions for PingFederate Bridge to validate whether one of the attribute values matches the specified value. When an attribute has multiple values, using a single-value condition causes the criteria to fail.

4. In the **Value** field, enter the comparison value.

> ⓘ **Note:**
>
> Values are compared verbatim. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions. For more information, see *Attribute mapping expressions* on page 37.

5. In the **Error Result** field, enter a custom error message.

Error results are handled differently for IdP-initiated single sign-on (SSO) and SP-initiated SSO requests.

IdP-initiated SSO

Redirect

When an InErrorResource URL is provided, the value of the **Error Result** field is used by an ErrorDetail query parameter in the redirect URL.

Template

When an InErrorResource URL is not provided, the value of the **Error Result** field is used by the variable *$errorDetail* in the `idp.sso.error.page.template.html` template file.

SP-Intiated SSO

SAML

The **Error Result** field value is used by the StatusMessage element in the response to the SP.

WS-Federation (Template)

> The **Error Result** field value is used by the *$errorDetail* variable in the `<pf_install>/`
> `pingfederate/server/default/conf/template/sourceid-wsfed-idp-`
> `exception-template.html` template file.

Using an error code in the **Error Result** field allows the error template or an application to process the code in a variety of ways. For example, the template or application can display an error message or e-mail an administrator.

To use localized descriptions, enter a unique alias in the **Error Result** field, such as `someIssuanceCriterionFailed`. Insert the same alias with the desired localized text in the applicable language resource files, located in the `<pf_install>/pingfederate/server/default/conf/language-packs` directory.

If not defined, PingFederate Bridge returns `ACCESS_DENIED` when the criterion fails at runtime.

6. Click **Add**.
7. Optional: Repeat to add more criteria.
8. If you require complex evaluations, including conditional criteria or partial matching, define them using attribute mapping expressions. For more information, see *Attribute mapping expressions* on page 37.

   a. Click **Show Advanced Criteria**.
   b. In the **Expression** field, enter the required expressions.
   c. Optional: In the **Error Result** field, enter an error code or message.

   > ⓘ **Note:**
   >
   > If the expressions resolve to a string value instead of `true` or `false`, the returned value overrides the **Error Result** field value.

   d. Click **Add**.
   e. Optional: Click **Test**, enter values in the applicable fields, and verify the results.
   f. Optional: Repeat to add multiple criteria using attribute mapping expressions.

Reviewing the authentication source mapping
You can review and save your authentication source mapping in the **Summary** tab.

Steps

▪ To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
▪ To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

▪ To discard your changes, click **Cancel**.

Configuring protocol settings
The **Protocol Settings** tab provides the launching point for configuring partner endpoints, message customizations, and other protocol-specific settings for browser-based single sign-on (SSO) connections.

Before you begin

For initial steps to configure a service provider (SP) connection, see *Accessing SP connections* on page 63.

For initial steps to configure Browser SSO, see *Configure IdP Browser SSO* on page 69.

About this task

 SAML 2.0

- Outbound SSO bindings (POST, artifact) and the corresponding assertion consumer service (ACS) URLs
- Outbound SLO bindings (POST, redirect, artifact, SOAP) and the corresponding protocol endpoints
- Inbound bindings (POST, redirect, artifact, SOAP)
- Artifact lifetime
- Signature policy
- Encryption policy

 SAML 1.x

- Outbound SSO bindings (POST, artifact) and the corresponding assertion consumer service (ACS) URLs
- Default target URL
- Artifact lifetime
- Signature policy

 WS-Federation

- Protocol endpoint
- Default target URL

Steps

1. Before configuring Browser SSO protocol settings, you must first configure assertion configuration. For more information, see *Configuring SSO token creation* on page 72

2. In the **Protocol Settings** tab, click **Configure Protocol Settings** to begin.

Setting Assertion Consumer Service URLs (SAML)
If your PingFederate Bridge configuration uses any version of SAML, you can configure assertion indexes, bindings, and endpoint URLs on the **Assertion Consumer Service URL** tab.

Before you begin
For prerequisites and initial steps to configure Browser SSO protocol settings, see *Configuring protocol settings* on page 88.

About this task

The assertion consumer service (ACS) endpoint is a location to which the single sign-on (SSO) tokens are sent, according to partner requirements. ACS is applicable to all SAML versions and both the identity provider (IdP)- and service provider (SP)-initiated SSO profiles.

ⓘ **Note:**

The SP might request that the SAML assertion be sent to one of several URLs, using different bindings. PingFederate Bridge uses the defined URL entries on this page to validate the authentication request. However, per SAML specifications, if the request is signed, PingFederate Bridge can verify the signature instead. The ACS URL does not necessarily need to be listed here. This is useful for scenarios where an ACS URL might be dynamically generated.

Some federation use cases might require additional customizations in the assertions sent from the PingFederate Bridge IdP server to the SP, such as placing well-formed XML in the <AttributeValue> element or including the optional SessionNotOnOrAfter attribute in the <AuthnStatement> element. You can use OGNL expressions to fulfill these use cases.

Steps

1. In the **Assertion Consumer Service URL** tab, configure one or more SAML ACS endpoints.
   a. Select a SAML binding from the **Binding**drop-down list.
   b. Enter the ACS endpoint URL to the **Endpoint URL** field.

      You can enter a relative path (begin with a forward slash) if you have provided a base URL on the **General Info** window.
   c. Optional: Select the **Default** box if you want this entry to be the default ACS endpoint.

      The administrative console always sets the first entry as the default ACS endpoint. You can reset the default endpoint when you add ACS endpoint.
   d. Optional: Enter an integer to the **Index** field for this ACS endpoint.

      The administrative console automatically assigns an index value for each ACS endpoint, starting from 0. If you want to define your own index values, you must make sure the index values are unique.
   e. Click **Add**.
   f. Optional: Repeat to add additional ACS endpoints.
2. Optional: Customize messages using OGNL expressions.

   ⓘ **Note:**

   OGNL expressions are not enabled by default. For more information about enabling and editing OGNL expressions, see *Attribute mapping expressions* on page 37.

   a. Click **Show Advanced Customizations**.
   b. Select a message type from the list.
   c. Enter an OGNL expression to fulfill your use case.

      ⓘ **Note:**

      For more information about **Message Type**, available variables, and sample OGNL expressions, see *Customizing assertions and authentication requests*.

   d. Click **Add**.
   e. Optional: Repeat to add another message customization.
3. Click **Next** to proceed to the next tab. For SAML 1.x configurations, see *Setting a default target URL (SAML 1.x)* on page 91. For SAML 2.0, see *Specifying SLO service URLs (SAML 2.0)* on page 93.

Results

If you are editing an existing connection, you can reconfigure any items, which could require additional configuration changes in subsequent tasks. You must always configure at least one ACS endpoint.

Setting a default target URL (SAML 1.x)

SAML 1.x service provider (SP) connections require that a default target URL be specified for a scenario where the identity provide (IdP) application does not include one in its single sign-on (SSO) request. This default URL represents the destination on the SP where the user will be directed.

Before you begin

For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

Steps

1. Enter default destination in the **Default Target URL** field.

2. Click **Next** to save your setting.

Results

If you are editing an existing connection, you update the target destination. You must always define a default destination when configuring a SAML 1.x SP connection.

Specifying the WS-Trust version

You can specify whether to use WS-Trust version 1.2 or 1.3 for tokens. The default version is 1.2.

Before you begin

For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

Steps

1. From the **WS-Trust Version** drop-down list, select version **1.2** or **1.3**.

    > ⓘ **Note:**
    >
    > For version 1.3, the response is always a `RequestSecurityTokenResponseCollection` object, as in the following example.
    >
    > ```
    > <wst:RequestSecurityTokenResonseCollection xmlns:wst="http://docs.oasis-
    > open.org/ws-sx/ws-trust/200512/">
    > ```
    >
    > For more information about WS-Trust version 1.3, see *OASIS WS-Trust 1.3 Standards*.

2. Click **Next** to save your changes.

Defining a service URL (WS-Federation)

On the **Service URL** tab, you can enter the WS-Federation protocol endpoint of your service provider (SP) partner where PingFederate Bridge will send single sign-on (SSO) tokens and single logout (SLO) cleanup messages.

Before you begin

For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

The SSO tokens are transmitted within a Request for Security Token Response (RSTR) message in response to a request for authentication from the SP. SLO cleanup messages are sent to your partner when PingFederate Bridge, as the identity provider (IdP), receives a user's SLO request. These cleanup messages indicate that the user's local session has been terminated.

To protect against session token hijacking, you can specify additional allowed domains and paths on this window. If the option to validate wreply for SLO is enabled, these additional domains and paths will also be taken into consideration as well. For more information, see *Managing partner redirect validation* on page 113.

Some federation use cases might require additional customizations in the RSTR message sent from the PingFederate Bridge IdP server to the SP. You can use OGNL expressions to fulfill these use cases.

Steps

1. On the **Service URL** tab, enter the WS-Federation protocol endpoint at the SP site in the **Endpoint URL** field.

   You can enter a relative path (begin with a forward slash) if you have provided a base URL on the **General Info** tab. For more information, see *Identifying the SP* on page 68.

2. Optional: Specify additional allowed domains and paths.

   a. Indicate whether to mandate secure connections when this resource is requested under **Require HTTPS**.

   > ⓘ **Important:**
   >
   > This selection is recommended to ensure that the validation will always prevent message interception for this type of potential attack, under all conceivable permutations.

   This check box is selected by default.

   b. Enter the expected domain name or IP address of this resource under **Valid Domain Name**.

   Enter a value without the protocol, such as `example.com` or `10.10.10.10`.

   Prefix a domain name with a wildcard followed by a period to include subdomains using one entry. For instance, `*.example.com` covers `hr.example.com` or `email.example.com` but not `example.com`, the parent domain.

   > ⓘ **Important:**
   >
   > While using an initial wildcard provides the convenience of allowing multiple subdomains using one entry, consider adding individual subdomains to limit the redirection to a list of known hosts.

   c. Optional: Enter the exact path of this resource under **Valid Path**.

   Start with a forward slash, without any wildcard characters in the path. If left blank, any path under the specified domain or IP address is allowed. This value is case-sensitive. For instance, `/inbound/Consumer.jsp` allows `/inbound/Consumer.jsp` but rejects `/inbound/consumer.jsp`.

   You can allow specific query parameters with or without a fragment by appending them to the path. For instance, `/inbound/Consumer.jsp?area=West&team=IT#ref1001` matches `/`

```
inbound/Consumer.jsp?area=West&team=IT#ref1001 but not /inbound/Consumer.jsp?
area=East&team=IT#ref1001.
```

    d. Optional: Select the check box under **Allow Any Query/Fragment** to allow any query parameters or fragment for this resource.

       Selecting this check box also means that no query parameter and fragment are allowed in the path defined under **Valid Path**.

       This check box is not selected by default.

    e. Click **Add**.

       Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Use the **Delete** and **Undelete** workflow to remove an existing entry or cancel the removal request.

    f. Repeat these steps to define multiple expected resources.

> ⓘ **Note:** The display order does not matter. A more specific match is considered a better match and an exact match is considered the best match.

3. Optional: Customize messages using OGNL expressions.

Expressions are not enabled by default. For more information about enabling and editing OGNL expressions, see *Attribute mapping expressions* on page 37.

    a. Click **Show Advanced Customizations**.

    b. Select a message type from the list.

    c. Enter an OGNL expression to fulfill your use case.

> ⓘ **Note:**
>
> For more information about **Message Type**, available variables, and sample OGNL expressions, see *Customizing assertions and authentication requests*.

    d. Click **Add**.

    e. Optional: Repeat to add another message customization.

4. Click **Next** to save your changes.

Results

If you are editing an existing connection, you can reconfigure any items, which might require additional configuration changes in subsequent tasks.

Specifying SLO service URLs (SAML 2.0)
On the **SLO Service URLs** tab, you associate bindings to the endpoints where your service provider (SP) receives logout requests when single logout (SLO) is initiated at your site and where PingFederate Bridge sends SLO responses when it receives SLO requests from the SP.

Before you begin
For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

This step applies only to SAML 2.0 connections when either SLO profile is selected on the **SAML Profiles** tab. For more information, see *Choosing SAML 2.0 profiles* on page 70.

Steps

1. Select a SAML binding from the list.

2. Enter the SLO endpoint URL to the **Endpoint URL** field.

   You can enter a relative path (begin with a forward slash) if you have provided a base URL on the **General Info** tab. For more information, see *Identifying the SP* on page 68.

3. Optional: Enter a URL in the **Response URL** field.

   When specified, this URL is the location to which SLO logout response messages are sent based on your partner agreement. When omitted, PingFederate Bridge sends logout responses to the SLO endpoint URL.

   You can enter a relative path (begin with a forward slash) if you have provided a base URL on the **General Info** window. For more information, see *Identifying the SP* on page 68.

4. Click **Add**.

5. Optional: Repeat to add additional SLO endpoints.

6. Click **Next** to save your settings.

Results

If you are editing an existing connection, you can reconfigure the SLO endpoints, which might require additional configuration changes in subsequent tasks.

Choosing allowable SAML bindings (SAML 2.0)
On the **Allowable SAML Bindings** tab, you select the one or more bindings that your service provider (SP) partner can use to send SAML authentication requests or single logout (SLO) messages.

Before you begin
For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

This step applies only to SAML 2.0 connections when the SP-initiated SSO profile or either SLO profile is selected on the **SAML Profiles** tab.

Steps

1. On the **Allowable SAML Bindings** tab, select the applicable SAML bindings based on your partner agreement.

   > ⓘ **Note:**
   >
   > If you have specified an Assertion Consumer Service (ACS) or SLO endpoint using the artifact (outbound) binding, you must including SOAP as one of the allowable (inbound) binding.

2. Click **Next** to save changes and proceed to **Artifact Resolver Locations**. For more information, see *Specifying artifact resolver locations (SAML 2.0)* on page 95.

Results

If you are editing an existing connection, you can reconfigure the allowable bindings, which might require additional configuration changes in subsequent tasks.

Setting an artifact lifetime (SAML)
When PingFederate Bridge sends an artifact to your service provider (SP)'s SAML ACS endpoint or SAML 2.0 SLO endpoint, an element in the message indicates how long it should be considered valid. On the **Artifact Lifetime** tab, you can specify the expiry information in seconds.

Before you begin
For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

You can change the default value to meet your requirements. You should also consider synchronizing your serve clock with your partner's SAML gateway server. If clocks are not synchronized, you might need to set the artifact lifetime to a higher value to prevent latency issues.

Steps

1. Optional: On the **Artifact Lifetime** tab, override the default value of the **Artifact Lifetime** field.

   The default value is 60 (seconds).
2. Click **Next** to save your changes.

Specifying artifact resolver locations (SAML 2.0)
When the artifact binding is enabled as one of the allowable bindings on the **Allowable SAML Bindings** tab, you must provide at least one artifact resolution service (ARS) endpoint on the **Artifact Resolver Locations** tab.

About this task

The ARD endpoint is where PingFederate Bridge sends back-channel requests to resolve artifacts received from the service provider (SP).

Steps

1. On the **Artifact Resolver Locations** tab, enter the ARS endpoint URL in the **URL** field.

   You can enter a relative path (begin with a forward slash) if you have provided a base URL on the **General Info** tab. For more information, see *Identifying the SP* on page 68.
2. Optional: Enter an integer to the **Index** field for this ACS endpoint.

   The administrative console automatically assigns an index value for each ARS endpoint, beginning with 0. If you want to define your own index values, you must make sure the index values are unique.
3. Click **Add**.
4. Optional: Repeat to add additional ARS endpoints.

   > ⓘ **Note:**
   >
   > When specifying multiple ARS endpoints, each endpoint must share the same transport protocol. That is, if one endpoint uses HTTPS, then all must use HTTPS.

5. After you have entered all of your ARS endpoints, click **Next** to save changes.

Results

If you are editing an existing connection, you can reconfigure any ARS endpoints.

Defining signature policy (SAML)

On the **Signature Policy** tab, you can control how digital signatures are used for SAML messages.

Before you begin

For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

The choices made in this tab depend on your partner agreement and your federation protocol. For more information, see *Digital signing policy coordination*.

SAML 2.0

Digital signing is required for SAML response messages sent from the identity provider (IdP) with the POST or redirect binding. Based on the SAML specifications, PingFederate Bridge provides three options:

- Select **Always Sign Assertion** to always sign the assertion portion inside the SAML response message.
- Select **Sign Response As Required** to sign the SAML response message per the SAML specifications. This is the default selection.
- Select both to always sign the assertion portion inside the SAML response message for all bindings and to sign the SAML response message per the SAML specifications.

Authentication request messages from the service provider (SP) may also be signed to enforce security. This scenario applies only when the SP-initiated single sign-on (SSO) profile is enabled on the **SAML Profiles** tab. Select **Require Authn Requests to be Signed** to enforce this digital signature requirement. For more information, see *Choosing SAML 2.0 profiles* on page 70.

SAML 1.x

For SAML 1.0 and SAML 1.1, the assertion portion inside the SAML response message can be digitally signed.

- Select **Always Sign Assertion** to always sign the assertion portion inside the SAML response message.

Steps

1. On the **Signature Policy** tab, select the options based on your partner agreement and federation protocol.
2. Click **Next** to save changes.

Results

If you are editing an existing connection, you can reconfigure the digital signature policy, which might require additional configuration changes in subsequent tasks.

Configuring XML encryption policy (SAML 2.0)

For SAML 2.0 configurations, in addition to using signed assertions to ensure authenticity, you and your partner can also agree to encrypt all or part of an assertion to improve privacy. If so, you can configure these settings on the **Encryption Policy** tab.

Before you begin

For prerequisites and initial steps for configuring Browser SSO protocols, see *Configuring protocol settings* on page 88.

About this task

> ⓘ **Note:** For WS-Fed connections with SAML 2.0 assertions, you cannot encrypt the entire assertion.

Option	Name identifier (SAML_SUBJEC	Other attributes	Encrypt the SAML_SUBJECT in SLO messages to the SP	Allow encryption in SLO messages from the SP
None	No encryption.	No encryption.	No encryption.	No encryption.
The entire assertion	Encrypted.	Encrypted.	Available as an option.	Available as an option.
One or more attributes	Available as an option.	Available as an option.	Available as an option only if you select to encrypt the name identifier (SAML_SUBJECT).	Available as an option only if you select to encrypt the name identifier (SAML_SUBJECT).

Steps

1. Select the options based on your partner agreement.
2. Click **Next** to save changes.

Results

If you are editing an existing connection, you can reconfigure the XML encryption policy, which might require additional configuration changes in subsequent tasks.

Reviewing protocol settings
On the **Summary** tab, you can review and save your protocol settings.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
- To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

- To discard your changes, click **Cancel**.

Reviewing browser-based SSO settings
On the **Summary** tab, you can review and save your browser-based single sign-on (SSO) settings.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.

▪ To keep your changes, click **Done** and continue with the rest of the configuration.

ⓘ **Tip:**

When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

▪ To discard your changes, click **Cancel**.

Configuring credentials

The **Credentials** tab provides the launching point for configuring security requirements you might need, depending on the federation protocol you are using and the choices you make.

Steps

▪ To continue, click **Configure Credentials**.

See subsequent topics for configuration steps.

Configuring digital signature settings

Digital signing is required for browser-based single sign-on (SSO) tokens and single logout (SLO) messages sent through POST or redirect bindings.

About this task

Digital signing is also required for WS-Trust STS service provider (SP) connections, for signing the outbound SAML security tokens.

For browser-based SSO, digital signing is not always required for profiles using the artifact or SOAP bindings unless you chose to sign the SAML assertion on **Protocol Settings**# **Signature Policy**, or the artifact resolution messages on **Back-Channel Authentication**# **Outbound SOAP Authentication Type**.

If digital signing is not required, PingFederate Bridge does not show the **Digital Signature Settings** tab.

Steps

1. On the **Digital Signature Settings** tab, select the certificate that you will use to sign the SSO tokens and SLO messages for the SP.
2. Select a signing certificate from the **Signing Certificate** list.

If you have not yet created or imported your certificate into PingFederate Bridge, click **Manage Certificates**. For more information, see *Manage digital signing certificates and decryption keys* on page 101.

ⓘ **Note:**

For WS-Federation connections using JSON Web Tokens (JWTs), only EC and RSA certificates are supported. RSA certificates must have a minimum key size of 2,048 bits. The **Signing Certificate** list automatically filters out certificates that do not meet these requirements.

---

3. Optional: Select the **Include the certificate in the signature <KeyInfo> element** check box if you have agreed to send your public key with the message.

> ⓘ **Note:**
>
> For WS-Trust STS, the `<KeyInfo>` element in the SAML token includes a reference to the certificate rather than the full certificate by default unless this check box is checked.

> ⓘ **Note:**
>
> This step is not applicable to WS-Federation connections using JWTs.

Select the **Include the raw key in the signature <KeyValue> element** check box if your partner agreement requires it.

4. Optional: Select the signing algorithm from the list.

The default selection is **RSA SHA256** or **ECDSA SHA256**, depending on the **Key Algorithm** value of the selected digital signing certificate. Make a different selection if you and your partner have agreed to use a stronger algorithm.

Reviewing SP credential settings
You can review, modify, save, and discard changes to your service provider (SP) credential settings.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
- To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

- To discard your changes, click **Cancel**.

Reviewing SP connection settings

When you finish creating or modifying a service provider (SP) connection, you can review the connection settings and toggle the connection status.

About this task

On the **Activation & Summary** tab, you can review, amend, discard, or save your changes.

Steps

- To amend your configuration, click the corresponding tab title and then follow the steps to complete the task.
- To keep your changes, click **Save**.
- To discard your changes, click **Cancel**.

Results

> ⓘ **Important:**

When creating a new connection, the default connection status is **Enabled** when you reach the **Activation & Summary** tab.

Whether you choose to disable a new connection now or later, you must click **Save** on the **Activation & Summary** tab if you want to keep the new connection.

The **SSO Application Endpoint** provides a sample URL at the `/idp/startSSO.ping` application endpoint that webmasters or web application developers at your site can use to invoke single sign-on for the connection. For a list of supported parameters, see *Viewing IdP application endpoints*.

Importing a connection

You can import a connection in the **Import Connection** window.

About this task

> ⓘ **Note:**
>
> Prior to the import, you can modify the XML file to suit your needs. The XML file can also be imported to another PingFederate Bridge environment acting in the same federation role. The source and the target must run the same version of PingFederate Bridge.

Steps

1. Go to **Applications**# **Integration**# **SP Connections**.
2. Click **Import Connection**.
3. Click **Choose File** and browse to a connection XML file.
4. Select the **Allow Update** check box to overwrite an existing connection with the imported file.
5. Click **Import** and **Done**.

Security

On the **Security** tab, you can configure several features related to certificate and key management, and system integration.

These features include:

- Signing and decryption keys and certificates
- Trusted CAs
- SSL server certificates
- Certificate revocation checking
- Partner metadata URLs
- System keys
- Redirect validation
- Incoming proxy settings
- Service authentication

Certificate and key management

The PingFederate Bridge administrative console provides a suite of configuration wizards for administrators to manage keys and certificates.

Tasks include:

- Managing trusted certificate authorities (CAs)
- Managing server certificates for the administrative port and runtime ports
- Managing client certificates for mutual TLS authentication
- Managing signing and decryption keys and certificates
- Managing OAuth and OpenID Connect keys
- Managing certificates from partners
- Configuring certificate revocation settings
- Managing partner metadata URLs
- Rotating system keys

> ⓘ **Note:**
>
> For certificates that you own, you have two export options: certificate only or certificate and private key.
>
> - Certificate only - PingFederate Bridge exports in PEM format with the file extension `.pem`.
> - Certificate and private key - PingFederate Bridge exports in PKCS #12 format with the file extension `.p12.`
>
> For features that use a certificate that you own, you can either create a new certificate or import an existing PKCS #12 certificate file.
>
> For partner certificates, you can only export the certificate. PingFederate Bridge exports the partner certificate in PEM format. You can also import a partner certificate in PEM format.

You can configure PingFederate Bridge to use a hardware security module (HSM) for cryptographic material storage and operations. When configured, private keys and their corresponding certificate are stored on the HSM. Related signing and decryption operations are processed there for enhanced security. Leveraging an HSM can help achieve FIPS 140-2 compliance for your deployment.

> ⓘ **Note:**
>
> Management of keys and certificates is restricted to administrative users with the **Crypto Admin** administrative role (see *Administrative accounts* on page 155).

See subsequent topics for configuration steps.

Manage digital signing certificates and decryption keys

On **Security**# **Certificate & Key Management**# **Signing & Decryption Keys & Certificates**, you can create and maintain certificates and their respective key pairs for the purpose of signing outgoing requests, responses, assertions, and access tokens, and for the purpose of decryption.

Use separate certificates for signing and decryption.

After creating your certificates, if they remain as self-signed certificates, you can enable automatic certificate rotation. See *Certificate rotation*.

Manage trusted certificate authorities

On the **Trusted CAs** window, you can import, export, review, and remove certificate authorities (CAs).

You can import your federation partner's CA certificate or self-signed certificates into PingFederate Bridge's global trust list on **Security# Certificate & Key Management# Trusted CAs**. If the CA is not one of the major authorities, you might also need to import the certificate from the CA that signed the partner certificate.

> ⓘ **Note:**
>
> If a required CA certificate is already available from the Java runtime, you do not need to import the same certificate into the PingFederate Bridge store.

Manage SSL server certificates

On the **Security # Certificate & Key Management# SSL Server Certificates** window, you can establish and maintain the certificates presented for access to the PingFederate Bridge administrative console (or the administrative API) and for incoming HTTPS connections at runtime.

The first system-generated certificate is the default certificate for both the administrative console and the runtime server. As multiple certificates are created, you can activate or deactivate them for the administrative console, the runtime server, or both. Additionally, you can select any of them as the new default certificate for the administrative console, the runtime server, or both at a later time.

When creating a certificate, you can add additional domain names through the use of the **Subject Alternative Names** field. Furthermore, if a user agent includes the host name that it intends to reach as part of the TLS handshake, PingFederate Bridge selects the applicable certificate based on the provided Server Name Indication (SNI) information. The selection looks at the common name and subject alternative names of each activated certificate. If PingFederate Bridge finds no match, it serves the default certificate. If PingFederate Bridge finds multiple matches, it serves the certificate with the better match.

SSL Server Certificates configuration

Certificate	Common name	Subject alternative names	Activation status
#1	www.example.com	(None)	Administrative console and runtime server
#2	www.example.org	*.example.org and test.example.local	Administrative console and runtime server
#3	www.example.info	*.example.info and *.example.com	Administrative console and runtime server
#4	admin.example.local	(None)	Administrative console (Default) and runtime server
#5	runtime.example.local	(None)	Administrative console and runtime server (Default)

Runtime behavior

Request type	Host name from SNI	Certificate served
Administrative or runtime	www.example.com	The host name from the SNI is an exact match to the common name of certificate #1 and a partial match to the second subject alternative name (`*.example.org`) of certificate #3. An exact match is a better match, so PingFederate Bridge serves certificate #1.
Administrative or runtime	www.example.org	The host name from the SNI is an exact match to the common name of certificate #2. PingFederate Bridge serves certificate #2.
Administrative or runtime	sso.example.org	The host name from the SNI is a partial match to the first subject alternative name (`*.example.org`) of certificate #2. There is no other exact or partial match. PingFederate Bridge serves certificate #2.
Administrative or runtime	sso.example.info	The host name from the SNI is a partial match to the first subject alternative name (`*.example.info`) of certificate #3. There is no other exact or partial match. PingFederate Bridge serves certificate #3.
Administrative or runtime	sso.example.com	The host name from the SNI is a partial match to the second subject alternative names (`*.example.com`) of certificate #3. There is no other exact or partial match. PingFederate Bridge serves certificate #3.
Administrative	www.example.local	The host name from the SNI does not match any configured certificate. PingFederate Bridge serves certificate #4, the default certificate for the administrative console.
Runtime	localhost	The host name from the SNI does not match any configured certificate. PingFederate Bridge serves certificate #5, the default certificate for the runtime server.

ⓘ **Note:**

If PingFederate Bridge finds multiple certificates of the same matching quality, it returns one of them in the TLS handshake. This response should not impact the user agent because either the common name or one of the subject alternative names matches the host name of the request. If PingFederate Bridge should always serve a particular certificate for any given host name, ensure that the common name and any configured subject alternative names do not overlap among multiple certificates.

Creating a new certificate
On the **SSL Server Certificates** window, you can generate customized certificates.

Steps

1. On the **SSL Server Certificates** window, click **Create new**.
2. On the **Create Certificate** window, enter the required information.

   For information about each field, refer to the following table.

Field	Description
Common Name	The common name (CN) identifying the certificate.
Subject Alternative Names	The additional DNS names or IP addresses that can be associated with the certificate.
Organization	The organization (O) or company name creating the certificate.
Organizational Unit	The specific unit within the organization (OU).
City	The city or other primary location (L) where the company operates.
State	The state (ST) or other political unit encompassing the location.
Country	The country (C) where the company is based.
Validity (days)	The time during which the certificate is valid.
Cryptographic Provider	The storage facility of the certificate. Applicable and visible only when PingFederate Bridge is integrated with an HSM in hybrid mode. ▪ Select **HSM** to store the certificate in the HSM. ▪ Select **Local Trust Store** to store the certificate in the local trust store managed by PingFederate Bridge.
Key Algorithm	A cryptographic formula used to generate a key. PingFederate Bridge uses either of two algorithms, RSA or EC.
Key Size (bits)	The number of bits used in the key. (RSA-1024, 2048 and 4096; and EC-256, 384 and 521.)
Signature Algorithm	The signing algorithm of the certificate. (RSA-SHA256, SHA384, and SHA512; and ECDSA-SHA256, SHA384, and SHA512.)

3. When finished, click **Next**.
4. On the **Summary** window, review your configuration, amend as needed, and click **Save**.

Importing a certificate and its private key
On the **SSL Server Certificates** window, you can import certificates.

Steps

1. On the **SSL Server Certificates** window, click **Import**.
2. On the **Import Certificate** window, choose the applicable certificate file and enter its password.

   ⓘ **Note:** If PingFederate Bridge is integrated with a hardware security module (HSM) from Thales, you cannot use an elliptic curve (EC) certificate as an SSL server certificate. You must select a certificate that uses the RSA key algorithm.

3. If PingFederate Bridge is integrated with an HSM in hybrid mode, select the storage facility of the certificate from the **Cryptographic Provider** list.
    a. Select **HSM** to store the certificate in the HSM.
    b. Select **Local Trust Store** to store the certificate in the local trust store managed by PingFederate Bridge.
4. On the **Summary** window, review your configuration, amend as needed, and click **Save**.

Creating a certificate-authority signing request (CSR)
On the **SSL Server Certificates** window, you can generate a CSR file for a certificate.

Steps

1. On the **SSL Server Certificates** window, select **Action**# **Certificate Signing** for the certificate.

> ⓘ **Note:**
>
> This selection is inactive if you have not yet saved a newly created or imported certificate. Click **Save** and then return to this window to initiate the process.
>
> The selection is also inactive if a previously signed certificate has been revoked. Because the revocation may indicate that the private key has been compromised, the best practice is to import or create a replacement certificate for certificate signing.

2. On the **Certificate Signing** window, select the **Generate CSR** option.
3. On the **Generate CSR** window, click **Export** to save the CSR file, and then click **Done**.

> ⓘ **Note:**
>
> Once saved, you can submit this CSR file to a certificate authority (CA) for a CA-signed certificate.

Importing a certificate-authority response (CSR response)
On the **SSL Server Certificates** window, you can import CSR response files for certificates.

Steps

1. On the **SSL Server Certificates** window, select **Action**# **Certificate Signing** for the certificate.
2. On the **Certificate Signing** tab, select the **Import CSR Response** option. Click **Next**.
3. On the **Import CSR Response** tab, click **Choose File**, and select the applicable CSR response file. Click **Next**.
4. On the **Summary** tab, review your configuration, and click **Save**.

Exporting a certificate
On the **SSL Server Certificates** window, you can export a certificate with or without its private key.

Steps

1. On the **SSL Server Certificates** window, select **Action**# **Export** for the certificate.

2. On the **Export Certificate** window, select the export type.
   a. Select **Certificate Only** to export the selected certificate without its private key. This is the default choice.
   b. Select **Certificate and Private Key** to export the selected certificate with its private key.

> ⓘ **Note:**
>
> This export contains the private key of the certificate. You must also enter an encryption password.
>
> If the selected certificate is stored in a hardware security module (HSM), the **Certificate and Private Key** option does not apply.

3. On the **Export & Summary** window, click **Export** to save the certificate file, and then click **Done**.

Reviewing a certificate
On the **SSL Server Certificates** window, you can review a particular certificate.

Steps

1. On the **SSL Server Certificates** window, select the certificate by its serial number.
2. Review the selected certificate in the pop-up window.
3. When finished, close the pop-up window.

Activating or deactivating a certificate
On the **SSL Server Certificates**, you can configure whether to activate or deactivate a certificate.

Steps

1. On the **SSL Server Certificates** window, select the relevant option under **Action** for the certificate.

   Any certificate can be activated for the administrative console, the runtime server, or both. When multiple certificates are activated for the administrative console (or the runtime server), you can deactivate any of them as long as one certificate remains active. Additionally, you may select any of them as the default certificate.
2. Click **Save** to keep your configuration.

Removing a certificate
On the **SSL Server Certificates** window, you can delete unwanted certificates.

Steps

1. On the **SSL Server Certificates** window, select **Action**# **Delete** for the certificate.

> ⓘ **Note:**
>
> If the selected certificate is activated for the administrative port, the runtime port, or both, the **Delete** option does not apply.
>
> To cancel the removal request, select **Action**# **Undelete** for the certificate.

2. Click **Save** to confirm your action.

Configuring certificate revocation

You can choose whether to use certificate revocation list (CRL) checking or Online Certificate Status Protocol (OCSP) checking as your preferred verification method.

About this task

By default at runtime, whenever a certificate revocation list (CRL) distribution-point URL is included within the certificate, PingFederate Bridge attempts to retrieve a CRL to verify that a signing certificate is not revoked. Optionally, on **Security**# **Certificate & Key Management**# **Certificate Revocation Checking**, you can enable and configure OCSP checking as the preferred verification method, depending on your requirements. For more information, see *Certificate validation*.

You can use OCSP in place of CRL checking, or retain CRLs as a backup method for failover.

> ⓘ **Note:**
>
> When OCSP is enabled, CRL checking is not done independently, but only as a failover option for one or more OCSP failure conditions.

Steps

1. Optional: Configure OCSP.

For more information about each field, see the following table.

Field	Description
Enable OCSP	Turns on OCSP certificate-revocation checking. OCSP checking is not enabled by default.
Default OCSP Responder URL	The location of a URL to use for certificate-revocation checking, a backup used only if the OCSP Responder URL is not contained in the certificate.
Default OCSP Responder Signature Verification Certificate	Certificate used to verify that the returned certificate status was sent from the Default OCSP Responder—required if the certificate is not included in the response. Click **Manage Certificates** to import the verification certificate.
Do NOT allow Responder to use cached responses	When not selected, the OCSP Responder uses cached responses when available for the subject certificate for an indicated period of time—see the description for **Next Update Grace Period**. If checked, PingFederate Bridge sends a nonce in the request to the Responder, effectively requiring that the status of the certificate be determined in real time. This option is intended to enhance the prevention of Internet replay attacks (in addition to timestamping), where required. > ⓘ **Important:** > Making this selection might slow down OCSP response time for a request and will increase general processing overhead at the Responder site. This check box is not selected by default.

Field	Description
This Update Grace Period (min)	To consider the response valid, the PingFederate Bridge server-clock time must correspond to the `<thisUpdate>` timestamp in the OCSP response, plus or minus the number of minutes set for this field to compensate for clock variances.
	The default value is `5` minutes.
Next Update Grace Period (min)	If the response includes a `<nextUpdate>` timestamp indicating when updated certificate statuses are available, then PingFederate Bridge checks to ensure that the timestamp is not earlier than the current server time, adding this grace period to compensate for clock variances.
	The default value is `5` minutes.
Responder Timeout (sec)	The allowable response time before the OCSP Responder URL is considered unavailable and processing continues. See the **OCSP Responder is Unavailable** setting.
	The default value is `5` seconds.
Response Caching Interval (hrs)	The number of hours that PingFederate Bridge caches the OCSP response.
	The default value is `48` hours.
Certificate is Unknown	The certificate does not fall under the purview of the certificate authority (CA) associated with the OCSP Responder. The choices indicate whether an unknown certificate is considered valid, or whether to try CRL checking.
	The default selection is **Treat as Revoked**.
OCSP Responder is Unavailable	Indicates what action to take if you cannot reach the Responder. The choices indicate whether an unknown certificate is considered valid, or whether to try CRL checking.
	The default selection is **Treat as Valid**.
OCSP Responder Returns Error	Indicates what action to take if the Responder returns an error. The choices indicate whether an unknown certificate is considered valid, or whether to try CRL checking.
	The default selection is **Treat as Revoked**.

2. Optional: Configure CRL checking.

For more information about each field, see the following table.

Field/Selection	Description
Enable CRL Checking	Enables CRL revocation checking.
	ⓘ **Note:**
	CRL checking must remain enabled if any selections for OCSP Error Handling include failover. If OCSP is enabled and no CRL failover is specified, then this selection has no effect.
	CRL revocation checking is enabled by default.

Field/Selection	Description
Treat Unretrievable CRLs as Revoked	If selected, PingFederate Bridge immediately aborts the processing associated with the certificate.
	If not selected, the server treats the certificate as valid but continues trying to retrieve the CRL.
	This check box is not selected by default.
Next Retry on Resolution Failure (min)	Specifies the number of minutes the server waits before trying to retrieve a CRL when the previous attempt failed—applies only when **Treat Unretrievable CRLs as Revoked** is unchecked.
	The default value is `1440` minutes, which is 24 hours.
Next Retry on Next Update Expiration (min)	How long the server waits before requesting a new CRL when the most recently retrieved CRL (in cache) has a next-update time in the past.
	ⓘ **Note:** Certain actions in the administrative console, such as saving changes to an identity provider (IdP) adapter instance, reset the CRL cache. When this happens, PingFederate Bridge requests new CRLs for subsequent transactions as needed.
	The default value is `60` minutes.
Verify CRL Signature	When selected (recommended), PingFederate Bridge verifies the CRL signature using the public key of the issuer, which must be in the certificate chain or in the list of Trusted CAs.
	This check box is selected by default.
Proxy Settings	If CRL checking is routed through a proxy server, specify the server's host DNS name or IP address and the port number. The same proxy information applies to OCSP checking, when enabled.

Manage Partner metadata URLs

On the **Security**# **Certificate & Key Management**# **Partner Metadata URLs** window, you can add, update, review, or remove SAML metadata URLs provided by your partners.

SAML metadata URLs streamline the process of establishing and maintaining SAML connections. If your partner provides SAML metadata by URL, you can use the metadata URL for the following scenarios:

- Creating a new SAML connection using the metadata URL and associating the metadata URL with the new connection
- Enabling or disabling automatic updates from the associated metadata URL
- Adding or updating the metadata URL associated with an existing SAML connection
- Updating an existing SAML connection using the metadata URL instantly

ⓘ **Tip:**

You can quickly create connections with InCommon participants, update the connections automatically or manually as the InCommon participants update their metadata, and do so securely knowing PingFederate Bridge only commits changes to your connections after validating the digital signatures of the signed metadata.

When PingFederate Bridge accesses a digitally signed metadata URL for the first time, it validates the digital signature and stores the metadata URL and its verification certificate if the signature is correct. When an existing metadata URL is accessed, PingFederate Bridge validates the digital signature using the stored certificate. If there is a digital signature error, PingFederate Bridge aborts the process and provides an error with a recommended course of action. You can bypass the signature verification process.

Rotating system keys

On the **System Keys** window, you can manually rotate your PingFederate system keys to optimize your environment's security.

About this task

System keys are used in cryptographic operations to generate and consume internal tokens. These tokens are leveraged in multiple use cases such as one-time links for self-service password reset and email ownership verification. Periodic rotation ensures optimal security of your environment.

Steps

1. Go to **Security# Certificate & Key Management# System Keys**.
2. To rotate the system keys, click **Rotate**.
3. Click **Save**.

Results
PingFederate generates a new **Pending** key. The key that was **Pending** becomes the **Current** key. The key that was **Current** becomes the **Previous** key.

System integration

You can configure PingFederate Bridge to act in accordance with your circumstances.

If PingFederate Bridge acts in a service provider (SP) role, you can configure redirect validation rules to ensure valuable information, such as user attribute values, is sent only to a list of designated target resources. If PingFederate Bridge is deployed behind a reverse proxy or a load balancer, you can configure whether and how PingFederate Bridge should extract contextual information from the requests. You can manage the availability and authentication requirements for the supporting services that PingFederate Bridge offers. You can find these menu items in the **Security** tab under the **System Integration** section.

Configuring redirect validation

Ensure that a designated target exists by validating single sign-on (SSO), single logout (SLO) and self-service user account management transactions.

About this task

You can configure several service provider (SP) adapters to pass security tokens or other user credentials from the PingFederate Bridge SP server to the target resource via HTTP query parameters, cookies, or POST transmittal. In all cases, these transport methods carry the risk that a third party (with specific knowledge of the identity provider (IdP), the SP, or both, PingFederate Bridge endpoints, and PingFederate Bridge configuration) could obtain and use valid security tokens to gain improper access to the target resource.

This potential security threat involves using a well-formed SSO or SLO link to start an SSO or SLO request for a resource at the SP site. However, the target resource designated in the link intercepts the security token by a redirection to a malicious website. This same threat also applies to self-service user account management endpoints when such requests include the TargetResource parameter.

To prevent such an attack, PingFederate Bridge provides a means of validating SSO, SLO, and self-service user account management transactions to ensure that the designated target resource exists through a list of configurable URLs. At minimum, an expected resource requires a domain name (or an IP address) and the selection of one or more applicable request types.

> ⓘ **Note:**
>
> The following default target URLs are always allowed, and you don't need to enter them into the list manually:
>
> - The default target URL for any IdP connections (see *Configuring default target URLs*)
> - The default target URL for any adapter-to-adapter mappings (see *Configuring a default target URL (optional)*)
> - The SP default URL for successful SSO (see *Configuring default URLs*)
> - The IdP default URL for successful SLO (see *Configuring a default URL and error message* on page 39)

PingFederate Bridge can also validate the error resource parameter. For more information about the InErrorResource parameter, see *IdP endpoints*, *SP endpoints* and *System-services endpoints*.

> ⓘ **Important:**
>
> PingFederate Bridge enables both target resource validation and error resource validation by default in new installations.
>
> For backward compatibility, PingFederate Bridge upgrade tools do not enable these options if they aren't selected in the previous PingFederate Bridge installation. Although optional, we strongly recommend enabling validation for both target and error resources and entering all expected resources (including the HTTPS option) to prevent unauthorized access.

Steps

1. Go to **Security**# **System Integration**# **Redirect Validation**.
2. Configure target resource validation options.

Option	Description
SSO	When selected, PingFederate Bridge validates the requested target resource for IdP connections, adapter-to-adapter mappings, and SAML 2.0 IdP Discovery against a list of configurable resources.
	This check box is selected by default in new installations.
	Clear the check box to disable the feature.
SLO and Other	When selected, PingFederate Bridge validates the requested target resource for SLO and self-service user account management requests against a list of configurable resources.
	This check box is selected by default in new installations.
	Clear the check box to disable the feature.

3. Configure error resource validation.

> ⓘ **Note:**
>
> Select the **Enable InErrorResource Validation** check box to validate the requested InErrorResource parameter value against a list of configurable resources.
>
> This check box is selected by default in new installations.
>
> Clear the check box to disable the feature.

4. Define a list of expected resources.

    a. Indicate whether to mandate secure connections when this resource is requested under **Require HTTPS**.

> ⓘ **Important:**
>
> This selection is recommended to ensure that the validation will always prevent message interception for this type of potential attack, under all conceivable permutations.

    This check box is selected by default.

    b. Enter the expected domain name or IP address of this resource under **Valid Domain Name**.

    Enter a value without the protocol, such as `example.com` or `10.10.10.10`.

    Prefix a domain name with a wildcard followed by a period to include subdomains using one entry. For instance, `*.example.com` covers `hr.example.com` or `email.example.com` but not `example.com`, the parent domain.

> ⓘ **Important:**
>
> While using an initial wildcard provides the convenience of allowing multiple subdomains using one entry, consider adding individual subdomains to limit the redirection to a list of known hosts.

    c. Optional: Enter the exact path of this resource under **Valid Path**.

    Start with a forward slash, without any wildcard characters in the path. If left blank, any path under the specified domain or IP address is allowed. This value is case-sensitive. For instance, `/inbound/Consumer.jsp` allows `/inbound/Consumer.jsp` but rejects `/inbound/consumer.jsp`.

    You can allow specific query parameters with or without a fragment by appending them to the path. For instance, `/inbound/Consumer.jsp?area=West&team=IT#ref1001` matches `/`

`inbound/Consumer.jsp?area=West&team=IT#ref1001` **but not** `/inbound/Consumer.jsp?`
`area=East&team=IT#ref1001.`

    d. Optional: Select the check box under **Allow Any Query/Fragment** to allow any query parameters or
fragment for this resource.

Selecting this check box also means that no query parameter and fragment are allowed in the path
defined under **Valid Path**.

This check box is not selected by default.

    e. Select one or more request types for this resource.

- Select the check box under **TargetResource for SSO** if this is an expected SSO target resource
for one or more IdP connections, adapter-to-adapter mappings, or SAML 2.0 IdP Discovery.
- Select the check box under **TargetResource for SLO and Other** if this is an expected target
resource for SLO and self-service user account management requests.
- Select the check box under **InErrorResource** if this is an expected InErrorResource parameter
value.

These check boxes are not selected by default.

    f. Click **Add**.

Use the **Edit**, **Update**, and **Cancel** workflow to make or undo a change to an existing entry. Use the
Delete and **Undelete** workflow to remove an existing entry or cancel the removal request.

    g. Repeat these steps to define multiple expected resources.

> ⓘ **Note:** The display order does not matter. A more specific match is considered a better match
> and an exact match is considered the best match.

5. Click **Save**.

Managing partner redirect validation
PingFederate Bridge enables you to validate a parameter for single logout (SLO) in order to prevent
unauthorized access.

About this task

Some of the parameters used to perform redirection represent locations at a partner site—for example, the
wreply parameter in WS-Federation. To protect against session token hijacking through open redirections,
PingFederate Bridge provides an option to validate wreply for single logout (SLO). Once enabled,
the parameter value is managed within the connection on a per-partner basis. PingFederate Bridge
amalgamates the entries from all active WS-Federation connections and validates wreply against the
consolidated list.

> ⓘ **Important:**
>
> PingFederate Bridge enables wreply validation for SLO by default in new installations.
>
> For backward compatibility, PingFederate Bridge upgrade tools do not enable this option if it was not
> selected in the previous PingFederate Bridge installation. Although optional, enabling wreply validation
> for SLO and specifying the allowed domains and paths for each WS-Federation connection can prevent
> unauthorized access.

Steps

1. Go to **Security**# **Redirect Validation**# **Partner Redirect Validation**.

2. Select the **Enable wreply Validation For SLO** check box to enable this feature.

> ⓘ **Note:**
> This check box is selected by default in new installations. Clear the check box to disable the feature.

3. Click **Save**.

Configure incoming proxy settings

Use the options in the **Incoming Proxy Settings** window to enable PingFederate Bridge to access information to construct correct responses to incoming requests.

When PingFederate Bridge is deployed behind a reverse proxy (or a similar network traffic management solution, such as a load-balancer), the following options enable PingFederate Bridge to use information in HTTP headers added by the reverse proxy to construct correct responses. These options, configurable on **Security# System Integration# Incoming Proxy Settings**, apply globally to all incoming requests.

HTTP header for client IP addresses

The **HTTP Header for Client IP Addresses** field allows you to globally specify the header name (for example, `X-Forwarded-For`) where PingFederate Bridge should attempt to retrieve the client IP address in all HTTP requests sent to PingFederate Bridge. Defining this field helps PingFederate Bridge identify the correct client IP address when PingFederate Bridge is operating behind a reverse proxy or load balancer.

Proxies commonly append the IP address from an incoming request to the `X-Forwarded-For` (or similar) header. If you enter `X-Forwarded-For` as the value of the **HTTP Header for Client IP Addresses** field, PingFederate Bridge combines multiple comma-separated header values in the same order that they are received. Define which IP address you want to use in the list box:

- Leave the default of **Use Last Value** to use the last value in the combined list.
- Select **Use First Value** to use the first value in the combined list.

HTTP header for hostname

The **HTTP Header for Hostname** field allows you to globally specify the header name (for example, `X-Forwarded-Host`) where PingFederate Bridge should attempt to retrieve the hostname and port in all HTTP requests sent to PingFederate Bridge. Proxies commonly append the hostname and port from an incoming request to the `X-Forwarded-Host` (or similar) header. If you enter `X-Forwarded-Host` as the value of the **HTTP Header for Hostname** field, PingFederate Bridge combines multiple comma-separated header values into the same order that they are received. Define which hostname you want to use in the list box:

- Leave the default of **Use Last Value** to use the last value in the combined list.
- Select **Use First Value** to use the first value in the combined list.

Client certificate header name and chain header name

If you use mutual client certificate authentication and want to use the Apache HTTP Server with `mod_ssl` as the incoming proxy, configure the Apache HTTP Server to pass client certificates as HTTP request headers and enter the header names on the **Incoming Proxy Settings** window.

The following examples shows the Apache HTTP Server configured to pass the client leaf certificate and up to four intermediate certificates as headers.

```
...
SSLOptions +ExportCertData
RequestHeader set LEAF_CERT "%{SSL_CLIENT_CERT}s"
RequestHeader set CHAIN0    "%{SSL_CLIENT_CERT_CHAIN_0}s"
RequestHeader set CHAIN1    "%{SSL_CLIENT_CERT_CHAIN_1}s"
```

```
RequestHeader set CHAIN2     "%{SSL_CLIENT_CERT_CHAIN_2}s"
RequestHeader set CHAIN3     "%{SSL_CLIENT_CERT_CHAIN_3}s"
...
```

ⓘ **Note:**

This configuration snippet is for demonstration purposes only.

To configure PingFederate Bridge to consume these HTTP request headers for the purpose of mutual client certificate authentication:

- Enter `LEAF_CERT` as the **Client Certificate Header Name**.
- Enter `CHAIN` as the **Client Certificate Chain Header Name**.

ⓘ **Note:**

Do not enter the trailing number from the chain header names.

ⓘ **CAUTION**:

Since HTTP request headers could potentially be forged, you should only specify a **Client Certificate Header Name** and a **Client Certificate Chain Header Name** if the Apache HTTP Server is immediately in front of your PingFederate Bridge environment. The specified values must match the header names used in the Apache HTTP Server configuration, omitting the trailing number from the chain header names.

Incoming proxy terminates HTTPS connections

The **Incoming proxy terminates HTTPS connections** option allows you to globally specify that connections to the reverse proxy are made over HTTPS even when HTTP is used between the reverse proxy and PingFederate Bridge.

Configuring service authentication

Administrators with the **Admin** role can activate and configure authentication for Attribute Query, Java Management Extensions (JMX), and SSO Directory Service.

About this task

If you are using the SAML 2.0 Attribute Query profile as a service provider (SP), then the requesting applications at your site must authenticate to the PingFederate Bridge server. For more information, see *Attribute Query and XASP* on page 116 and the */sp/startAttributeQuery.ping* SP application endpoint.

Authentication is required to access PingFederate Bridge runtime data via JMX (see *Runtime monitoring using JMX* on page 117) or to make SOAP calls to the Connection Management Service. Authentication is optional for the SSO Directory Service. For more information, see *Web service interfaces and APIs* and *SSO Directory Service*.

ⓘ **Note:**

To help ensure network security, access to all of these services is deactivated when PingFederate Bridge is first installed.

To activate and configure authentication for the Connection Management Service, grant the administrators all three administrative roles: **Admin**, **Crypto**, and **User Admin**. For more information, see *Connection Management Service* on page 118.

Steps

- To enable a service:
  a. On **Security**# **System Integration**# **Service Authentication**, select **Action**# **Activate** for your desired service.
  b. Enter or modify) the service account **ID** and define or reset the **Shared Secret**.

     You and the application developer must agree to these values.

  > ⓘ **Tip:**
  >
  > Authentication is optional for the SSO Directory Service.

- To disable a service, on **Security**# **Service Authentication**, select **Deactivate** under **Action** for your desired service.

  > ⓘ **Note:**
  >
  > Although not accessible when deactivated, the Connection Management Service and the SSO Directory Service are deployed by default with PingFederate Bridge. If your organization does not plan to use one or both of these services, you can remove the following WAR file or files:
  >
  > - `<pf_install>`/pingfederate/server/deploy2/pf-mgmt-ws.war for the Connection Management Service
  > - `<pf_install>`/pingfederate/server/deploy/pf-ws.war for the SSO Directory Service

Attribute Query and XASP

The SAML 2.0 Attribute Query profile allows a service provider (SP) to request user attributes from an identity provider (IdP) in a secure transaction separate from single sign-on (SSO).The X.509 Attribute Sharing Profile (XASP) defines a specialized extension of the general Attribute Query profile.

The IdP, acting as an attribute authority, accepts attribute queries, performs a datastore lookup into a user repository such as an LDAP directory, provides values to the requested attributes, and generates an attribute response back to the originating SP requester. The SP then returns the attributes to the requesting application.

> ⓘ **Tip:**
>
> When privacy is required for sensitive attributes, you can configure PingFederate Bridge to obfuscate, or mask, their values in the server and transaction logs.

Web SSO is distinct from the Attribute Query use case. You can configure PingFederate Bridge servers to implement either of these profiles without regard to the other.

The XASP specification enables organizations with an investment in Public Key Infrastructure (PKI) to issue and receive Attribute Queries based on user-certificate authentication.

Under XASP a user authenticates directly with an SP application by providing their X.509 certificate. Once the user is authenticated, the SP application requests additional user attributes by contacting the SP PingFederate Bridge server. A portion of the user's X.509 certificate is included in the request and can be used to determine the correct IdP to use as the source of the requested attributes. Finally, the SP generates an Attribute Query and transmits it to the IdP over the SOAP back channel.

Because the user arrives at the SP server already authenticated, no PingFederate Bridge adapter is used in this case.

Runtime monitoring using JMX

PingFederate Bridge supports runtime monitoring and reporting through Java Management Extensions (JMX). JMX technology represents a Java-centric approach to application management and monitoring.

JMX exposes instrumented code in the form of MBeans. Application management systems that support JMX technology, such as JConsole, can request runtime information from the PingFederate Bridge JMX server.

ⓘ **Important:**

Authentication is required for JMX-client access to PingFederate Bridge runtime data. For more information, see *Configuring service authentication* on page 115.

ⓘ **Tip:**

You can use HTTP requests at any time to verify the status of the PingFederate Bridge server. For more information, see *Customizing the heartbeat message*.

You can also supplement monitoring information by applying third-party analysis and reporting tools to the security audit log, in which PingFederate Bridge records fine-grain details, including response times and event types, for all server transactions. For more information, see *Security audit logging*.

PingFederate Bridge JMX server reports monitoring data for single sign-on (SSO) and single logout (SLO) transactions. In addition, numerous Jetty-standard MBeans are available to the PingFederate Bridge server's JMX clients.

SSO and SLO monitoring

For SSO/SLO transaction processing, PingFederate Bridge provides these MBeans:

- `pingfederate:type=TOTAL_FAILED_TRANSACTIONS`
- `pingfederate:type=TOTAL_TRANSACTIONS`

PingFederate resets these counters to zero after restart.

Sample Jetty metrics

The following table describes examples of Jetty MBean metrics, available through JMX, that administrators might find useful to supplement information provided through the PingFederate Bridge-specific MBeans.

MBean	Attributes
`org.eclipse.jetty.server: connectorstatistics`	`connections` – The total number of TCP connections accepted by the server.
For Jetty connectors including the primary and secondary PingFederate Bridge runtime server ports.	`connectionsDuration`* – How long connections are kept open. Maximum, mean, standard deviation, and total accumulated time are available.
	`connectionsOpen` – The current number of open connections. Maximum is also available (`connectionsOpenMax`).

MBean	Attributes
`org.eclipse.jetty.server.handler:` `statisticshandler`	`requests` – Total number of requests received.
	`requestsActive` – Number of requests currently being processed. Max is also available.
	`requestTime` – Request duration. Maximum, mean, standard deviation, and total accumulated time are available.
	`responses1xx, responses2xx, responses3xx, …` – Total number of requests that returned HTTP status codes of 1xx, 2xx, 3xx, etc.
`org.eclipse.jetty.util.thread:` `queuedthreadpool` Two pools: one for the runtime server, with 200 maximum threads; one for the administrative console, with 20 maximum threads.	`idleThreads` – Number of idle threads currently available.
	`threads` – Number of threads currently running, including both idle and active.
	`minThreads` – Minimum number of threads in the pool.
	`maxThreads` – Maximum number of threads in the pool.
	`lowOnThreads` – A boolean flag indicating whether the pool is running low on threads.

`java.lang: Memory`
`java.lang: MemoryPool`
`java.lang: GarbageCollection`
`java.lang: OperatingSystem`
	Various attributes measuring CPU usage and memory.

Advanced JMX configuration

PingFederate Bridge uses port 1099 for its JMX server. To change the port or other Java Message Service (JMS) configuration items, if needed, modify the `jmx-remote-config.xml` configuration file in the `<pf_install>`/pingfederate/server/default/conf directory.

> ⓘ **Note:**
>
> When connecting to the JMX service using SSL, the default, ensure that the client trusts the PingFederate Bridge SSL server certificate presented. For more information, see *Manage SSL server certificates* on page 102.

Connection Management Service

The Connection Management Service supports basic connection management capabilities and is accessible only on a PingFederate Bridge server running the administrative console.

> ⓘ **Note:**
>
> As of PingFederate 10.2, the Connection Management Service has been deprecated and will be removed in a future release.

The Connection Management Service is useful in a variety of circumstances. Consider the following use cases:

- Using the Connection Management Service as a utility, you can migrate changes to a partner connection through staging environments. For example, development, test, and production.

  Using the Connection Management Service, you might need to make changes to URLs and keys to make the connection appropriate to the next environment.
- Using the Connection Management Service, an external application can update or delete connections programmatically, or create new ones using an exported connection XML file as a template.

You can find the WAR file for this service, `pf-mgmt-ws.war`, in the `<pf_install>`/pingfederate/ `server/default/deploy2` directory.

> ⓘ **Note:**
>
> If you do not want to allow use of the service, do not deploy it: remove the WAR file from the `deploy2` directory.

The SOAP-accessible service endpoint is `pf-mgmt-ws/ws/ConnectionMigrationMgr`.

The web services Description Language (WSDL) document describing this service can be retrieved from `/ pf-mgmt-ws/ws/ConnectionMigrationMgr?wsdl`.

System

On the **System** tab, you can configure several features related to data and credential stores, the server, and external systems.

These features include:

- *Data & Credential Stores*
- *Server*
- *External Systems*
- *Configuring the Active Directory environment*

Data & Credential Stores

Under **Data & Credential Stores**, you can configure datastores, password credential validators, and Active Directory domains and Kerberos realms.

See the following sections:

- *Managing datastores* on page 120
- *Password Credential Validators* on page 134
- *Configuring the Active Directory environment*

Managing datastores

On the **Data Stores** window, you can create, modify, review, and remove datastores as needed.

Steps

1. Go to **System**# **Data & Credential Stores**# **Data Stores**.

   - To create a new datastore, click **Add New Data Store** and then follow the configuration wizard to complete the task.
   - To modify an existing datastore, select the datastore and then follow the configuration wizard to complete the task.
   - To review usage of an existing datastore, click **Check Usage** under **Action**.
   - To remove an existing datastore or cancel the removal request, click **Delete** or **Undelete** under **Action**.

   > ⓘ **Note:**
   >
   > You can only remove datastores that are not currently in use.

   - To fine-tune the caching interval for datastore validation, update the **Data-Store Validation Interval** field value to the desired amount of time in seconds.

   > ⓘ **Note:**
   >
   > As you configure various components on the administrative console, PingFederate Bridge performs connectivity tests against the applicable datastores. By default, PingFederate Bridge stores successful test results for five minutes. This design improves the performance of the administrative console by reducing the number of calls it makes to the target servers and the amount of time it takes to move from one configuration window to another.

   > ⓘ **Note:**
   >
   > The default value is `300` seconds (five minutes). A value of `0` turns off the caching and validation tests are executed with each access. This setting applies to all datastores.

2. To keep your changes, click **Save**. To discard your changes, click **Cancel**.

Adding a new datastore

On the **Data Stores** window, you can create and configure a new datastore.

Steps

1. Go to **System**# **Data & Credential Stores**# **Data Stores**.
2. Click **Add New Data Store**.
3. Enter a name for the datastore.
4. From the **Type** list, select the type of datastore.

   Available types are limited to the ones currently installed on your server.
5. Optional: To mask attribute values returned from this datastore in PingFederate Bridge logs, select the **Mask Values in Log** check box.
6. Click **Next**.

Configuring a JDBC connection

You can establish a Java Database Connectivity (JDBC) connection to your database server.

About this task

> ⓘ **Note:**
>
> PingFederate Bridge was tested with vendor-specific JDBC 4.2 drivers. For more information, see
> *Database driver information* on page 11. To obtain the database driver JAR file, contact your database
> vendor. Install the database driver file to the `<pf_install>/pingfederate/server/default/lib`
> directory then restart the server.

Steps

1. Go to **System**# **Data & Credential Stores**# **Data Stores**.
2. On the **Data Stores** window, click **Add New Data Store**.
3. On the **Data Store Type** tab, type a name for the datastore.
4. From the **Type** list, select **Database (JDBC)**. Click **Next**.
5. Optional: To mask attribute values returned from this datastore in PingFederate Bridge logs, select the **Mask Values in Log** check box.
6. Click **Next**.
7. In the **Database Config** window, configure your JDBC connection. Information about each field is provided in the following table.

Field	Description
JDBC URL	The location of the database server and the database. The structure of the JDBC URL varies depending on the vendor. You can add multiple JDBC URLs. You can also specify which node is the default by clicking **Set as Default** under **Action**. > ⓘ **Tip:** > For Oracle MySQL, to enable automatic reconnection attempts when the connection is not available at runtime, enter a SQL statement in the **Validate Connection SQL** field and add the following query string to the JDBC URL: > `?autoReconnect=true`
Tags	Tags are defined in the `node.tags` property in the `<pf_install>/pingfederate/bin/run.properties` file. For a description of the `node.tags` property, see *Deploying cluster servers* on page 165. In PingFederate Bridge deployments that are regional, you can enter one or more tags for a JDBC URL, which specifies with which datastore that particular PingFederate Bridge node should communicate. If none of the tags match what is defined for the `node.tags` property, the default node is used. The following rules apply to tags: ▪ You must separate multiple tags specified for one node with spaces. ▪ You can't use a tag more than once per datastore. ▪ Tags are optional. If needed, you can configure a non-default node without tags. Doing this is useful if you are not yet ready to tag the

Field	Description
	node, or if you are still in the planning stage but want to enter the address for the node now.
Driver Class	The name of the driver class used to communicate with the source database. The driver class name should be supplied by the database software vendor in a JAR file.
Username	The name that identifies the user when connecting to the database.
Password	The password needed to access the database.
Validate Connection SQL **(Optional but recommended)**	A simple SQL statement used by the PingFederate Bridge runtime server to verify that the database connection is still active and to reconnect if needed. If a SQL statement is not provided here, PingFederate Bridge might not reconnect to the database if the connection is broken. ⓘ **Important:** Ensure that the SQL statement is valid for your database. For example: ▪ `SELECT 1 from dual` (for Oracle Database or Oracle MySQL) ▪ `SELECT getdate()` (for Microsoft SQL Server) ▪ `SELECT 1` (for PostgreSQL) ⓘ **Tip:** To use this feature for Oracle MySQL, you must also add the `?autoReconnect=true` query parameter to the JDBC URL.
Mask Values in Log	Determines whether all attribute values returned through this datastore should be masked in PingFederate Bridge logs. Applicable only when editing an existing datastore.
Allow Multi-Value Attributes	When selected, indicates that the JDBC datastore can select more than one record from a column and return the results as a multivalued attribute. Otherwise, a query returns only the first value in the column.

8. Click **Test Connection** to determine whether the administrative node can communicate with the specified datastore.

ⓘ **Note:**

Datastore validation is no longer enabled during configuration. This feature lets you configure datastores without requiring a successful connection between the administrative node and the datastore. You can also save the datastore even if the connection is not currently successful.

9. Click **Advanced** to configure additional settings.

   a. On the **Advanced Database Options** window, click **Apply Defaults** to view or restore default values.

   > ⓘ **Tip:**
   >
   > The default values are conservative based on the server thread pool settings configured in the `<pf_install>`/pingfederate/etc/jetty-runtime.xml file. If any changes are made to thread pooling, we recommend updating settings as outlined in the next step.

   b. Configure advanced settings.

   For more information about each field, see the following table.

Field	Description
Minimum Pool Size	The smallest number of database connections that can remain in the pool for the given datastore. A minimum value of `0` means that the minimum number of connections in the pool is zero. > ⓘ **Note:** > > For optimal performance, the value for this setting should equal 50% of the maxThreads value in the Jetty server configuration (see *Configuring connection pools to datastores*). Note that PingFederate Bridge does not establish the connection pool for the given datastore until it receives a request that requires one or more attributes from that datastore. The default value (after clicking on **Apply Defaults**) is `10`.
Maximum Pool Size	The largest number of database connections that can remain in the pool for the given datastore. > ⓘ **Note:** > > For optimal performance, the value for this setting should equal 75% to 100% of the maxThreads value in the Jetty server configuration (see *Configuring connection pools to datastores*). The default value (after clicking on **Apply Defaults**) is `100`.
Blocking Timeout (ms)	The amount of time a request waits to get a connection from the connection pool before it fails. A value of `-1` means that a request waits indefinitely for the connection pool to return a connection. The default value (after clicking on **Apply Defaults**) is `5000`.
Idle Timeout (min)	The length of time the connections can sit idle in the pool before it closes them. A value of `-1` means that the connection pool does not close its connections (once established). Note that PingFederate Bridge maintains the minimum connection pool for the given datastore once the pool is established. The default value (after clicking on **Apply Defaults**) is `5`.

10. Click **Save** to save your configuration.

Configuring an LDAP connection
You can establish an LDAP connection to your directory server.

Steps

1. Go to **System**# **Data & Credential Stores**# **Data Stores**.
2. On the **Data Stores** window, click **Add New Data Store**.
3. On the **Data Store Type** tab, type a name for the datastore.
4. From the **Type** list, select **Directory (LDAP)**.
5. Optional: To mask attribute values returned from this datastore in PingFederate Bridge logs, select the **Mask Values in Log** check box.
6. Click **Next**.
7. On the **LDAP Configuration** tab, configure your LDAP connection as described in the following table.

Field	Description
Data Store Name	The name of the datastore.
	Applicable only when editing an existing datastore.
Hostname(s) (Required)	The network address of the directory server, either an IP address, a host name, or a fully qualified domain name. The entry might include a port number; for example, `10.10.10.101:1389`. For failover, enter multiple directory servers, each separated by a space. In addition to network error conditions, PingFederate Bridge also fails over to the next server if the current server returns an LDAP system error.
	ⓘ **Note:**
	If multiple directory servers are specified, each server must be accessible by using the same user distinguished name (DN) and password (unless the **Bind Anonymously** check box is selected).
	You can add multiple hostnames. You can also specify which node is the default by clicking **Set as Default** under **Action**.
	PingFederate Bridge can also leverage DNS service records to locate the directory server (when the **Use DNS SRV Record** check box is selected), in which case the value of this field must be a single domain; for example, `example.com`.
Tags	Tags are defined in the `node.tags` property in the `<pf_install>`/`pingfederate/bin/run.properties` file. See *Deploying cluster servers* on page 165 for a description of the `node.tags` property.
	In regional PingFederate Bridge deployments, you can enter one or more tags for a host name, which specify with which datastore that particular PingFederate Bridge node should communicate. If none of the tags match what is defined for the `node.tags` property, the default node is used.
	The following rules apply to tags:
	▪ You must separate multiple tags specified for one node with spaces.
	▪ You cannot use a tag more than once per datastore.
	▪ Tags are optional. If needed, you can configure a non-default node without tags. This is useful if you are not yet ready to tag the node, or if you are still in the planning stage but want to enter the address for the node now.

Field	Description
Use LDAPS	When selected, PingFederate Bridge connects to the directory server using LDAPS. This selection applies equally to all servers specified in the **Hostname(s)** field.
	ⓘ **Important:**
	We recommend securing all LDAP connections by using LDAPS.
	ⓘ **Note:**
	To enable the password changes, password reset, or account unlock features in the HTML Form Adapter against Microsoft Active Directory, you must secure the connection to your directory server using LDAPS; Microsoft Active Directory requires this level of security to allow password changes.
	This check box is not selected by default.
Use DNS SRV Record	Used in conjunction with the domain information defined in the **Hostname(s)** field and the preference of LDAP or LDAPS, PingFederate Bridge uses DNS SRV records to locate the directory server when this check box is selected. You can fine-tune the TTL value and the record prefixes on the **Advanced LDAP Options** window.
	ⓘ **Note:**
	When the DNS returns multiple SRV records, PingFederate Bridge uses the record with the lowest-numbered priority value and fails over to the record with the next lowest priority value. If multiple records share the same priority value, PingFederate Bridge uses the records with the highest-numbered weight value.
	PingFederate Bridge repeats this exercise until it establishes a connection or fails to connect to any directory server after taking all records into consideration.
	This check box is not selected by default.
Follow LDAP Referrals	Select this check box to let the datastore follow LDAP referrals on Microsoft Active Directory or Oracle Unified Directory
	ⓘ **Note:**
	PingFederate Bridge always follows LDAP referrals from PingDirectory based on the recommended PingDirectory configuration.

Field	Description
LDAP Type (Required)	If you are using this datastore for outbound provisioning and your directory server is PingDirectory, Microsoft Active Directory, or Oracle Unified Directory, select the applicable type from the list, such that PingFederate Bridge can pre-populate many provisioning settings on **Outbound Provisioning**# **Channel**# **Source Settings**. ⓘ **Tip:** If your directory server is not PingDirectory, Microsoft Active Directory, or Oracle Unified Directory, you can define a custom LDAP Type to streamline the outbound provisioning configuration. The LDAP type is also used to enable password-change messaging between Microsoft Active Directory and PingFederate Bridge when an HTML Form Adapter instance is used.
Bind Anonymously	Select this check box if your directory server supports anonymous binding and if no credentials are needed to access the directory server. When selected, user DN and password are not required. ⓘ **Tip:** For inbound provisioning, because PingFederate Bridge needs to manage local user records, your directory server might require a specific service account to handle the communication between PingFederate Bridge and the target directory server. If you choose an anonymous binding, ensure that this access level provides permission to search the directory for user-account information. This check box is not selected by default.
User DN	The user name credential required to access the directory server. ⓘ **Important:** The service account must have permission to search the directory for user-account information. If your use cases involve reading from the directory server without creating, updating, or deleting any records, consider using a service account with read-only access. For inbound provisioning, a service account with permission to create, read, update, and delete users and groups is required. When connecting to a Microsoft Active Directory server, enter a Microsoft Active Directory user account. Do not use a computer account. When connecting to PingDirectory or Oracle Unified Directory, configure proxied authorization for the service account on the directory server if you intend to enable self-service password reset in any HTML Form Adapter instances that use this datastore. For more information, see *Proxied authorization*.
Password	The password credential required to access the directory server.

Field	Description
Mask Values in Log	Determines whether all attribute values returned through this datastore should be masked in PingFederate Bridge logs.
	Applicable only when editing an existing datastore.

8. Click **Test Connection** to determine whether the administrative node can communicate with the specified datastore.

> ⓘ **Note:**
>
> Datastore validation is no longer enabled during configuration. This feature lets you configure datastores without requiring a successful connection between the administrative node and the datastore. You can also save the datastore even if the connection is not currently successful.

9. Optional: Click **Advanced**. If you choose an anonymous binding, configure additional settings in the **Advanced LDAP Options** window.
10. Click **Save**.

Setting advanced LDAP options
PingFederate Bridge enables you to customize the default settings of both the search pool and the bind pool for each LDAP datastore.

About this task

PingFederate Bridge maintains a search pool and a bind pool for each LDAP datastore for optimal performance. The search pool is for LDAP directory searches. The bind pool is for LDAP bind authentication purposes. Use the **Advanced LDAP Options** window to change default pool settings. These settings are applicable to both the search pool and the bind pool.

When configuring PingFederate Bridge to locate the directory server based on DNS SRV record, you can fine-tune the TTL value and the SRV record prefixes.

Steps

1. In the **Advanced LDAP Options** window, click **Apply Defaults** to view or restore default values.

> ⓘ **Tip:**
>
> The default values are conservative based on the server thread pool settings configured in the `<pf_install>`/pingfederate/etc/jetty-runtime.xml file. If any changes are made to thread pooling, update the settings as outlined in the following step.

2. Configure advanced settings. For more information about each field, see the following table.

Field	Description
Test Connection on Borrow	Indicates whether to validate objects before they are borrowed from the pool. This check box is not selected by default.
Test Connection on Return	Indicates whether to validate objects before they return to the pool. This check box is not selected by default.
Create New Connection If Necessary	Indicates whether you can create temporary connections when the **Maximum Connections** threshold is reached. Temporary connections are managed automatically.

Field	Description
	ⓘ **Note:** If disabled, when the **Maximum Connections** value is reached, subsequent requests relying on this LDAP datastore instance might fail.
	This check box is selected by defaul
Verify LDAPS Hostname	Indicates whether to verify that the host name of the directory server matches the subject (CN) or one of the subject alternative names (SANs) from the certificate.
	ⓘ **Important:** Verify the LDAPS host name for all LDAPS connections.
	This check box is selected by default.
Minimum Connections **(Required)**	The smallest number of connections that can remain in each pool. A minimum value of 1 creates two connections, one connection in the search pool and one connection in the bind pool. The default value is 10.
	ⓘ **Note:** For optimal performance, the value for this setting should equal 50% of the maxThreads value in the Jetty server configuration. For more information see *Configuring connection pools to datastores*.
	ⓘ **Note:** PingFederate Bridge does not establish the connection pool for the given datastore until it receives a request that requires one or more attributes from that datastore.
Maximum Connections **(Required)**	The largest number of active connections that can remain in each pool (not including the temporary connections that are managed automatically when the **Create New Connection If Necessary** check box is selected). The value must exceed or equal the **Minimum Connections** value.
	ⓘ **Note:** For optimal performance, the value for this setting should equal 75% to 100% of the maxThreads value in the Jetty server configuration. For more information, see *Configuring connection pools to datastores*.
	The default value is 100.

Field	Description
Maximum Wait (Milli) **(Required)**	The maximum number of milliseconds the pool waits for an available connection when trying to obtain a connection from the pool. A value of -1 causes the pool not to wait at all and to either create a new connection or produce an error (when no connections are available). The default value is -1.
Time Between Eviction (Milli) **(Required)**	The number of milliseconds between periodic background health checks against the available connections in this pool. A value of -1 disables the evictor. The default value is 60000.
Read Timeout (Milli) **(Required)**	The maximum number of milliseconds a connection waits for a response to return before producing an error. A value of -1 causes the connection to wait indefinitely. The default value is 3000.
Connection Timeout (Milli) **(Required)**	The maximum number of milliseconds that a connection attempt can continue before returning an error. A value of -1 causes the pool to wait indefinitely. The default value is 3000.
DNS TTL (Milli) **(Required)**	The amount of time in milliseconds that a previously obtained DNS SRV record remains valid. When this threshold is reached, PingFederate Bridge contacts the DNS for a new SRV record to locate the directory server. The default value is 60000.
LDAP DNS SRV Record prefix **(Required)**	The prefix that PingFederate Bridge uses in its DNS queries for SRV records to locate an LDAP-capable directory server. The default value is _ldap._tcp.
LDAPS DNS SRV Record prefix **(Required)**	The prefix that PingFederate Bridge uses in its DNS queries for SRV records to locate an LDAPS-capable directory server. The default value is _ldaps._tcp.

3. Optional: Click **Next** to specify LDAP binary attributes on the **LDAP Binary Attributes** tab.

4. Click **Save**.

Configuring other types of datastores

Besides connecting to a directory server using LDAP or a database server using Java Database Connection (JDBC), PingFederate can connect to other types of datastores, such as REST API-enabled data sources that return user attributes in JavaScript Object Notation (JSON).

See the following topics for configuration steps:

- *Configuring a REST API datastore* on page 130
- *Configuring a custom datastore*

Configuring a REST API datastore

To retrieve attribute data from a JSON-based REST API, you must first create a REST API datastore.

Steps

1. Go to **System**# **Data & Credential Stores**# **Data Stores**.

2. On the **Data Stores** window, click **Add New Data Store**.

3. On the **Data Store Type** tab, type a name for the datastore.

4. From the **Type** list, select **Rest API**.

5. Optional: To mask attribute values returned from this datastore in PingFederate Bridge logs, select the **Mask Values in Log** check box.

6. Click **Next**.

7. On the **Configure Data Store Instance** tab, click **Add a new row to 'Base URLS and Tags'**.

    a. Enter the **Base URL** of the data source offering REST API access to its data. You can enter multiple base URLs.

    b. Optional: Enter one or more tags per base URL.

        Tags are defined in the `node.tags` property in the `<pf_install>`/pingfederate/bin/`run.properties` file. For a description of the `node.tags` property, see *Deploying cluster servers* on page 165.

        In PingFederate Bridge deployments that are regional, you can enter one or more tags for a Base URL, which specifies the PingFederate Bridge node the datastore should communicate with. If none of the tags match what is defined for the `node.tags` property, the default node is used.

        The following rules apply to tags:

        -   You must separate multiple tags specified for one node with spaces.
        -   Tags must unique per base datastore.
        -   You cannot use a tag more than once per datastore.
        -   Tags are optional. If needed, you can configure a non-default node without tags. Doing this is useful if you are not yet ready to tag the node, or if you are still in the planning stage but want to enter the address for the node now.

    c. Click **Update** under **Action**.

    d. Select **Set as Default** under **Action** beside the Base URL and Tags that you want to use as the default. The first Base URL and Tags configured is set as the default automatically.

    ⓘ **Note:** If the data source exposes multiple paths or requires specific query parameters to retrieve user records, enter the base URL here and then specify the path and query parameters in the attribute source configuration.

    For more information, see *Specifying a resource path for a REST API datastore* on page 133.

8. If the data source requires specific HTTP request headers, click **Add a new row to 'HTTP Request Headers'**.

    ⓘ **Note:** If configured, PingFederate Bridge includes the configured HTTP request headers and their values when contacting the data source.

    a. Enter the applicable name and value under **Header Name** and **Header Value**.

    b. Click **Update** under **Action**.

    Repeat these steps to define additional HTTP request headers and their values.

9. Click **Add a new row to 'Attributes'** to define local attribute names and map them to the data returned by the data source.

   Map each attribute to a path representing an attribute in the JSON response. This path follows the syntax defined in the *JavaScript Object Notation (JSON) Pointer* specification at tools.ietf.org/html/rfc6901.

   You must define at least one attribute.

   a. Enter the **Local Attribute** name and **JSON Response Attribute Path**.
   b. Click **Update** under **Action**.

   Repeat these steps to define additional attributes.

   > (i) **Tip:**
   >
   > Define only the attributes required by other configuration items, such as contract fulfillment or token authorization. Provide meaningful attribute names so that you can easily recognize them at a later time.

10. Select one of the following authentication methods.

    - **None**

      PingFederate Bridge makes unauthenticated REST API requests to the data source. No credential information is required. This is the default setting.

    - **Basic Authentication**

      PingFederate Bridge authenticates via the HTTP Basic authentication scheme. Enter the required credentials in the **Username** and **Password** fields.

    - **OAuth 2.0 Bearer Token**

      PingFederate Bridge authenticates by presenting an OAuth 2.0 access token.

      In this scenario, PingFederate Bridge is an OAuth client, specifically a client that uses the client credential grant type to obtain access token from an authorization server and presents the access token to the data source for authentication.

      Enter the client credentials in the **Client ID** and **Client Secret** fields. Then enter the token endpoint URL at the authorization server and the applicable scope (or scopes) in the **OAuth Token Endpoint** and **OAuth Scope** fields.

11. If PingFederate Bridge should mask attribute values returned through this datastore in its log, select the **Mask Values in Log** check box.

    This check box is visible only when editing an existing datastore and is not checked by default.

12. Optional: Click **Show Advanced Fields** to configure additional settings.

    For more information, see the following table.

Field	Description
Enable HTTPS Hostname Verification	Indicates whether to verify that the hostname of the data source matches the subject (CN) or one of the subject alternative names (SANs) from the certificate. > (i) **Important:** > We recommend to verify hostname for all connections. This check box is selected by default.

Field	Description
Read Timeout (MS)	Defines the socket timeout in milliseconds.
	Enter 0 to set an infinite timeout.
	Enter a negative integer to use the default value set by the operating system.
	The default value is 10000 in milliseconds, which is 10 seconds.
Connection Timeout (MS)	Determines the timeout in milliseconds until a connection is established.
	Enter 0 to set an infinite timeout.
	Enter -1 to use the default value set by the operating system.
	The default value is 10000 in milliseconds, which is 10 seconds.
Max Payload Size (KB)	Defines the maximum allowed size in kilobytes (KB) of the returned JSON response payload.
	Enter 0 to configure an unrestricted payload size.
	The default value is 1024 in KB.
Retry Request	Determines whether to retry a user data retrieval request if the data source returns an HTTP status code found in the **Retry Error Codes**.
	This check box is selected by default.
Maximum Retries Limit	Defines the maximum number of retry attempts if the data source returns an HTTP status code found in the **Retry Error Codes**.
	The default value is 5.
Retry Error Codes	Enter a comma-separated list of HTTP status codes, for which if received from the data source, PingFederate Bridge might retry the request.
	For example, you can enter 429 for "Too Many Request" or 503 for "Service Unavailable".
	The default value is 429.
Test Connection URL	Determines the URL to which PingFederate Bridge sends GET requests to test the datastore connection on the **Actions** tab.
	When not specified (the default), PingFederate Bridge sends GET requests to the base URL of the datastore.

13. On the **Actions** tab, verify the datastore configuration.

   a. Click **Test Connection** to test the connectivity between PingFederate Bridge and the data source.

   The administrative console displays the results returned by the data source. The PingFederate Bridge server log may contain additional messages as well.

   b. Review the results.
   c. Optional: Click **Reset** and repeat the test.

14. On the **Summary** tab, review your configuration, amend as needed, click **Save** to keep your configuration or click **Cancel** to discard it.

You have two use cases that can leverage user attributes obtained through REST APIs. The data source returns user records in JSON

```
{
  "uid": "asmith",
  "office": {
    "city": "Denver",
    "state": "CO",
    "zipCode": 80202
  },
  "telephoneNumbers": [
    "+1 303-555-1234",
    "+1 303-555-5678"
  ],
  "department": "Engineering"
}
```

The first use case requires the user's department, while the second use case requires the first telephone number and the ZIP code.

To address both use cases, create a REST API datastore with the following attributes.

Local Attribute	JSON Response Attribute Path
Dept	/department
Telephone	/telephoneNumbers/0
Zip	/office/zipCode

Once set up, you can fulfill various contracts or configure issuance criteria based on the attribute data from the data source.

Specifying a resource path for a REST API datastore
On the **Configure Data Source Filters** window, you can specify a resource path for a REST API datastore.

Steps

On the **Configure Data Source Filters** window, if the REST API datastore requires a relative path or additional query parameters, or both, to retrieve user records, enter them in the **Resource Path** field.

Example

You have use cases that can leverage user attributes obtained through REST APIs. The data source returns user records in JSON. It also provides the following paths to access its data based on user populations:

- https://rest.example.com/development/users
- https://rest.example.com/staging/users

To retrieve the record of a particular user, the request must include the uid query parameter with the identifier of the user.

Your use cases focus on users under the /staging/users path. Your authentication policy uses the HTML Form Adapter, which captures user identifiers using the username attribute.

To address this sample use case:

1. Create a REST API datastore with a base URL of https://rest.example.com.
2. Add the REST API datastore as an attribute source in the applicable use cases, such as a service provider (SP) connection that uses an HTML Form Adapter instance or an OAuth identity provider (IdP)

Adapter Mapping configuration that maps from an HTML Form Adapter instance into the persistent grants.

3. When prompted to configure the filtering option on the **Configure Data Source Filters** window, enter `/ staging/users?uid=${username}` in the **Resource Path** field.

Reviewing datastore configuration

On the **Summary** tab, you can review your data store configuration settings and then save them.

Steps

1. In the **Data Store Configuration** window, on the **Summary** tab, review your changes.

   - ▪ To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
   - ▪ To keep your changes, click **Done** and continue with the rest of the configuration.

   ⓘ **Tip:**

   When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

   - ▪ To discard your changes, click **Cancel**.

2. Click **Done**.

Password Credential Validators

PingFederate Bridge provides an authentication mechanism using plugin password credential validators (PCVs). This feature provides centralized credential validation for various PingFederate Bridge components and configurations.

To manage Password Credential Validators, go to **System**# **Data & Credential Stores**# **Password Credential Validators**.

For each instance of the HTML Form Adapter, the HTTP Basic Adapter, and the Username Token Processor, you can select the same PCV instance, a unique PCV instance, or multiple PCV instances. When you select multiple PCV instances for a given adapter or token processor instance, if the first PCV instance fails to authenticate a user, the PCV returns control to the adapter or the token processor. The adapter or the token processor then tries the next PCV instance. The cycle stops until a PCV instance succeeds or the last PCV instance also fails.

For OAuth clients using the Resource Owner Password Credentials grant type, you configure a grant-mapping configuration to fulfill the persistent grant contract using the attribute values from the applicable PCV instances.

ⓘ **Note:**

You can only create one grant-mapping configuration per applicable PCV instance.

If you want to manage OAuth client records using the OAuth Client Management Service or persistent grants using the OAuth Access Grant Management Service, you must select a PCV instance when configuring authorization server settings. When accessing these services, you must include in the requests valid credentials via HTTP Basic authentication scheme.

PingFederate Bridge is distributed with the following plugin PCVs.

LDAP Username Password Credential Validator

   Validates credentials based on an LDAP look-up in an organization's user-datastore.

PingID PCV (with integrated RADIUS server)

Validates credentials from a VPN RADIUS client based on an LDAP look-up in an organization's user-datastore. For more information, see *Integrate PingID with your VPN*.

PingOne for Enterprise Directory Password Credential Validator

Validates credentials stored in PingOne for Enterprise Directory.

RADIUS Username Password Credential Validator

Validates credentials based on the RADIUS protocol on an organization's RADIUS server.

Simple Username Password Credential Validator

Validates credentials maintained by PingFederate Bridge.

ⓘ **Note:**

By default, PingFederate Bridge automatically checks multi-connection errors whenever you access this window. This verifies that configured connections are not adversely affected by changes made here.

If you experience noticeable delays in accessing this page, you can disable automatic connection validation. Go to **System**# **Server**# **General Settings**.

Choosing a Password Credential Validator

You can choose the type of Password Credential Validator (PCV) that you want to use in PingFederate Bridge. You must also specify the PCV's name, ID, and whether it uses a parent instance.

About this task

Available PCV types are determined by plug-in `.jar` files loaded in the `<pf_install>`/`pingfederate/` `server/default/deploy` directory. Several validator plugins are bundled with PingFederate Bridge. You can add other plugins from the Ping Identity Downloads website.

Steps

1. On the **Type** tab, enter a name and an ID for the instance.
2. Select the type of the PCV from the **Type** list.
3. Optional: Select a **Parent Instance** from the list.

    Use this option when creating an instance that is similar to an existing one. The child instance inherits the configuration of its parent. You can also override one or more settings during the setup. Select the **Override ...** check box and make the adjustments as needed in one or more subsequent windows.

Password Credential Validator instance configurations

The instance configuration of a Password Credential Validator (PCV) varies depending on the credential validators deployed on your server.

For PCVs bundled with PingFederate, see the following topics:

-   *Configuring the LDAP Username Password Credential Validator* on page 136
-   *Configuring the PingOne for Enterprise Directory Password Credential Validator* on page 140
-   *Configuring the RADIUS Username Password Credential Validator* on page 141
-   *Configuring the Simple Username Password Credential Validator* on page 143

Configuring the LDAP Username Password Credential Validator

The LDAP Username Password Credential Validator (PCV) verifies credentials using an organization's LDAP datastore.

About this task

When an authentication error occurs, PingFederate Bridge automatically parses the messages returned by PingDirectory, Microsoft Active Directory (AD), Oracle Unified Directory (OUD), or Oracle Directory Server (ODS) and categorizes them with error conditions.

When validating against a directory server other than PingDirectory, AD, OUD, or ODS, administrators can define custom message categorization by mapping specific error messages with wildcard support to the desired error conditions.

The error messages are returned to the HTML Form Adapter instances and the OAuth clients using the Resource Owner Password Credential grant type. The HTML Form Adapter is designed to show the error message it receives from the LDAP Username PCV. OAuth-client developers can create custom experiences based on the error responses, which contain the error messages. The HTML Form Adapter uses the relevant error conditions to determine the LDAP password-change scenarios and to present the relevant messages to the end users.

> ⓘ **Tip:**
>
> These customizable messages are stored in the PingFederate Bridge message file, `pingfederate-messages.properties`, located in the `<pf_install>/pingfederate/server/default/conf/language-packs` directory.
>
> You can localize these messages by using the PingFederate Bridge localization framework for an international audience. For more information, see *Localizing messages for end users*.

Steps

1. Go to the **Instance Configuration** tab.
2. Optional: Override the authentication error messages.

   > ⓘ **Note:**
   >
   > You might require this option in order for a directory server other than PingDirectory, AD, OUD, or ODS to support the password change function in the HTML Form Adapter or to alter the end-user messages associated with that function.

   a. Click **Add a new row to 'Authentication Error Overrides'**.
   b. Enter an applicable LDAP error message under **Match Expression**.

      > ⓘ **Tip:**
      >
      > You can use wildcard asterisks to match messages returned from your directory server. For example, `*expired*`.

   c. Select a relevant error condition from the **Error** list.
   d. Optional: Enter a key name or an error message under **Message Properties Key**.

      No value

      > If you skip this field, PingFederate Bridge returns the default message based on the selected error condition.

A unique key name

If you enter a key name in this field and then add the key name with a key value (the desired error message) to the PingFederate Bridge message file, PingFederate Bridge returns that key value.

The key name must be unique. Furthermore, you may localize these messages by using the PingFederate Bridge localization framework for an international audience.

An error message

If you enter an error message in this field (without defining it in the PingFederate Bridge message file), PingFederate Bridge returns your message verbatim.

e. Click **Update** under **Action**.
f. Repeat these steps to add more overrides as needed.

> ⓘ **Note:**
>
> Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.
>
> Use the up and down arrows to change the display order. The display order does not affect runtime processing.

3. Select the LDAP datastore and enter information into the required fields.

For more information about each field, see the following table.

Field	Description
LDAP Datastore (Required)	The LDAP datastore configured in PingFederate Bridge. If you have not yet configured the server to communicate with the directory server you need, click **Manage Data Stores**. ⓘ **Note:** When connecting to an AD LDAP server, if you want to enable the password changes, password reset, or account unlock features in the HTML Form Adapter, you must secure the datastore connection to your AD LDAP server using LDAPS. AD requires this level of security to allow password changes.
Search Base (Required)	The location in the directory server from which the search begins.

Field	Description
Search Filter (Required)	The LDAP query to locate a user record.
	If your use case requires the flexibility of allowing users to identify themselves using different attributes, you can include these attributes in your query. For instance, the following search filter allows users to sign on using either the sAMAccountName or employeeNumber attribute value through the HTML Form Adapter.
	`(|(sAMAccountName=${username})(employeeNumber=${username}))`

> ⓘ **Important:**
>
> To ensure that your service providers (SPs) always get the expected attribute, select a specific user attribute as the source of the subject identifier when configuring the applicable SP connections. There are several ways to do so:
>
> - Extend the PCV contract and fulfill the subject identifier through the HTML Form Adapter. For more information, see *Extending the contract for the credential validator* on page 143.
> - Add a data source in the SP connection and fulfill the subject identifier through a datastore query. For more information, see *Configuring attribute sources and user lookup* on page 81.
> - If you use authentication policy in conjunction with a policy contract, you can add a data source in the contract mapping configuration and fulfill the subject identifier in an SP connection through the authentication policy contract. For more information, see *Applying policy contracts or identity profiles to authentication policies*.
>
> When configuring multifactor authentication using PingID, where you chain an instance of the PingID Adapter behind an HTML Form Adapter instance, ensure that you also select a specific user attribute as the incoming user attribute for the PingID Adapter instance. For example, if you have set up PingFederate Bridge as the identity bridge for your PingOne for Enterprise account and have selected sAMAccountName as the subject identifier in the SP connection, you should also select sAMAccountName as the incoming user attribute for your PingID Adapter instance. You can accomplish this through an instance of the Composite Adapter or an authentication policy.
>
> For more information, see **Input User ID Mapping** in *Configuring a Composite Adapter instance* or **Incoming User ID** in *Specifying incoming user IDs*, respectively.

Field	Description
Scope of Search	The level of search to perform in the search base.
	One Level indicates a search of objects immediately subordinate to the base object, not including the base object itself. **Subtree** indicates a search of the base object and the entire subtree within the base object distinguished name.
	The deault selection is **Subtree**.

Field	Description
Case-Sensitive Matching	The option to enable case-sensitive matching between the LDAP error messages returned from the directory server and the **Match Expression** values specified on this window.
	This check box is selected by default.

Advanced fields for self-service password reset, account unlock, and user name recovery through the HTML Form Adapter

Field	Description
Display Name Attribute	The LDAP attribute used for personalizing messages to the users.
	This field is applicable for all password reset types (other than **None**), account unlock, and user name recovery.
	The default value is `displayName`.
Mail Attribute (for password reset)	The LDAP attribute containing the email address of the users.
	This field is required when password reset using one-time link or one-time password is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.
	ⓘ **Note:**
	When configuring in conjunction with user name recovery, this attribute should correspond to the attribute specified on the left side of the **Mail Search Filter** field.
	The default value is `mail`.
SMS Attribute (for password reset)	The LDAP attribute containing the telephone number of the users.
	This field is required when password reset using text message is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.
	This field has no default value.
PingID Username Attribute (for password reset)	The LDAP attribute containing the PingID user name of the users.
	This field is required when password reset using **PingID** is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.
Mail Search Filter (for user name recovery)	The LDAP query to locate a user record using an email address, such as `mail=${mail}`.
	This field is required when user name recovery is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.
	ⓘ **Note:**
	When configuring in conjunction with password reset, the attribute specified on the left side of this search filter should correspond to the attribute specified in the **Mail Attribute** field.

Field	Description
Username Attribute (for user name recovery)	The LDAP attribute containing the user identifier of the users.
	This field is required when user name recovery is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.
	ⓘ **Note:** This attribute should correspond to the attribute specified on the left side of the **Search Filter** field.
Mail Verified Attribute (for user name recovery)	The LDAP attribute indicating whether the user's email address is verified. The expected value of this user attribute is either `true` or `false` (case insensitive).
	This field is required when user name recovery using only verified email addresses is enabled in any HTML Form Adapter instances that validate credentials against this LDAP Username PCV instance.

Configuring the PingOne for Enterprise Directory Password Credential Validator
The PingOne for Enterprise Directory Username Password Credential Validator (PCV) verifies credentials stored in your PingOne for Enterprise Directory.

Before you begin

To use the PingOne for Enterprise Directory PCV, you must have:

- A PingOne for Enterprise account
- A PingFederate Bridge account

For more information, see *Managing PingOne Directory Users* in the *PingOne for Enterprise Administration Guide*.

Steps

On the **Instance Configuration** tab, enter your account information in **Client ID** and **Client Secret**.
For more information about each field, refer to the following table. All fields are required. nti

Field	Description
Client ID	The REST API client ID is a unique identifier PingFederate Bridge uses to identify itself to the PingOne for Enterprise Directory API.
	For more information, see *View or renew directory API credentials* in the *PingOne for Enterprise Administration Guide*.
Client Secret	The client secret is used to authenticate the client ID against the PingOne for Enterprise Directory API.
	For more information, see *View or renew directory API credentials* in the *PingOne for Enterprise Administration Guide*.
Advanced Fields	
PingOne URL	The PingOne for Enterprise Directory API.
	The default value is https://directory-api.pingone.com/api.

Field	Description
Authenticate by Subject URL	The relative path for user authentication.
	The default value is `/directory/users/authenticate?by=subject`.
Reset Password URL	The relative path for password reset.
	The default value is `/directory/users/password-reset`.
SCIM User URL	The relative path for searching users requesting password reset.
	The default value is `/directory/user`.
Connection Pool Size	The maximum size of the connection pool to PingOne for Enterprise Directory.
	The default value is `100`.
Connection Pool Idle Timeout	The maximum time (in milliseconds) that a connection can remain idle before it is closed and removed from the connection pool.
	The default value is `4000`.

Configuring the RADIUS Username Password Credential Validator

The RADIUS Username Password Credential Validator verifies credentials using the RADIUS protocol.

About this task

RADIUS supports strong authentication with both one-step (a combination of regular password and a one-time password in one field) and two-step (challenge-response) authentication. Two-step authentication is supported in the HTML Form Adapter.

> ⓘ **Important:**
>
> If your RADIUS server is a Microsoft Network Policy Server (NPS), passwords containing special characters will not be encoded and decoded properly due to limitations with NPS.

> ⓘ **Tip:**
>
> RADIUS server messages are used by the HTML Form Adapter to determine the two-step authentication scenarios and to present a sign on window to the end users.

Steps

1. On the **Instance Configuration** tab, configure one or more RADIUS servers.
   a. Click **Add a new row to 'RADIUS Servers'**.
   b. In each field, enter the required information.

   For more information about each field, refer to the following table. All fields are required.

Field	Description
Hostname	The IP address of the RADIUS server.
	For failover, enter one or more backup RADIUS servers by adding each server in its own row of the table. Each row represents a distinct RADIUS server that can be used for failover. PingFederate Bridge attempts to make a connection to each server in the order listed until a successful connection is obtained.

Field	Description
Authentication Port	The UDP port used to authenticate to the RADIUS server.
	The default value is `1812`.
Authentication Protocol	The protocol used to authenticate to the RADIUS server.
	The available choices are Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Select the protocol expected by your RADIUS server.
	The default selection is **PAP**.
Shared Secret	The password shared between PingFederate Bridge and the RADIUS server used to encrypt the attribute identifying the NAS (Network Access Server) originating the request for access.

> ⓘ **Note:**
>
> The NAS-IP-Address attribute is added to all Access-Request packets sent to the RADIUS server. The value is copied from the pf.engine.bind.address property in the `<pf_install>/pingfederate/bin/run.properties` file. Only IPv4 addresses are supported.

c. Click **Update** in the **Action** column.

d. Repeat these steps to add more RADIUS servers as needed.

> ⓘ **Note:**
>
> Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.
>
> Use the up and down arrows to adjust the order in which you want PingFederate Bridge to attempt credential authentication. If an earlier RADIUS server fails to validate the credentials, PingFederate Bridge moves sequentially through the list until credential validation succeeds. If none of the RADIUS servers is able to authenticate the user's credentials, the credential validation process fails.

2. Optional: Click **Show Advanced Fields** to reconfigure default settings.

For more information about each field, refer to the following table. All fields are required.

Field	Description
NAS Identifier	The password shared between PingFederate Bridge and the RADIUS server used to encrypt the attribute identifying the NAS (Network Access Server) originating the request for access.
	The default value is `PingFederate`.
Timeout	The maximum number of milliseconds before a connection timeout to the RADIUS server.
	The default value is `3000`.
Retry Count	The number of times to retry a failed connection before moving to the next host.
	The default value is `3`.

Configuring the Simple Username Password Credential Validator
The Simple Username Password Credential Validator verifies credentials maintained by PingFederate Bridge. This validator is best used for testing purposes or for an organization with few accounts.

Steps

1. On the **Instance Configuration** tab, click **Add a new row to 'Users'**.
2. Enter a user name, followed by a password (twice).
3. Click **Update** in the **Action** column.
4. Repeat these steps to add more user credentials as needed.

> ⓘ **Note:**
>
> Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.
>
> Use the up and down arrows to adjust the order in which you want PingFederate Bridge to attempt credential authentication. PingFederate Bridge moves sequentially through the list until credential validation succeeds or no match is found.

Extending the contract for the credential validator
You can extend Password Credential Validator (PCV) instance contracts to return attribute values relevant to authenticated users.

About this task

In some use cases, you might want to extend the contracts of the PCV instance. For example, you might use extended attributes to map into a USER_KEY for an OAuth persistent grant configuration.

This capability allows the validator to return attribute values pertaining to the authenticated users from PingOne for Enterprise Directory, a directory server, or a RADIUS server.

> ⓘ **Tip:**
>
> If you are configuring an HTML Form Adapter instance with an instance of the LDAP Username Password Credential Validator, extend the contract of the adapter by the same attribute names in order for the credential validator to pass extended attribute values to the HTML Form Adapter instance.
>
> If you are configuring the HTML Form Adapter instance with an instance of the RADIUS Username Password Credential Validator, you only need to extend the contract of the HTML Form Adapter instance itself.

Steps

1. Copy the vendor-specific attribute dictionaries into the `pingfederate/server/default/conf/radius` directory.

> ⓘ **Note:**
>
> The format of the dictionaries must use the *FreeRadius dictionary syntax*.

2. Edit the existing `dictionary` file to include each of the dictonaries.

3. Optional: On the **Extended Contract** tab, enter an attribute name and click **Add**.

> ⓘ **Note:**
>
> Click **Edit**, **Update**, or **Cancel** to make or undo a change to an existing entry. Click **Delete** or **Undelete** to remove an existing entry or cancel the removal request.

Finishing the Password Credential Validator instance configuration
On the **Summary** tab, you can review your configuration.

Steps

- To keep your changes, click **Save**.
- To amend your configuration, click the name of the corresponding tab and then follow the configuration wizard to complete the task.
- To discard your changes, click **Cancel**.

Configuring the Active Directory environment

You can configure Active Directory to access a domain and enable Kerberos as an authentication option for it.

About this task

To enable Kerberos authentication, you must make several Active Directory configuration changes to grant PingFederate Bridge access to the domain and add the domain to PingFederate Bridge.

> ⓘ **Important:**
>
> Do not configure subdomains if the parent domain in the same forest is already configured. For more information, see *Multiple-domain support* on page 145.

> ⓘ **Note:**
>
> You must have Domain Administrator permissions to make the required changes.

Steps

1. Create a domain user account that PingFederate Bridge can use to contact the Kerberos Key Distribution Center (KDC). The account should belong to the Domain Users group. We recommend that you set the password with no expiration.

2. Use the Windows utility `setspn` to register Service Principal Name (SPN) directory properties for the account by executing the following command on the domain controller.

`setspn -s HTTP/<pf-idp.domain.name> <pf-server-account-name>` , where *<pf-idp.domain.name>* is the canonical name of the PingFederate Bridge server and *<pf-server-account-name>* is the domain account you want to use for Kerberos authentication. For more information on canonical name, see *https://tools.ietf.org/html/rfc2181#section-10*.

> ⓘ **Note:**
>
> When executing the `setspn` command, you must capitalize `HTTP` and follow it with a forward slash (`/`).

3. Verify that the registration was successful by executing the following command.

   ```
   setspn -l <pf-server-account-name>
   ```

   This gives you a list of SPNs for the account. Verify that `HTTP/<pf-idp.domain.name>` is one of them.

> ⓘ **Note:**
>
> After making an SPN change, any authenticated end users must re-authenticate by closing the browser or signing off and back on before attempting single sign-on (SSO).

Configuring Active Directory domains or Kerberos realms

You can configure an Active Directory (AD) domain or Kerberos realm to authenticate users.

Steps

1. Go to **System**# **Data & Credential Stores**# **Active Directory Domains/Kerberos Realms**.
2. From the **Manage AD Domains/Kerberos Realms** window, configure the AD environment to integrate with PingFederate Bridge. For more information, see *Configuring the Active Directory environment* on page 144.
3. Click **Add Domain/Realm** to create an AD domain.

> ⓘ **Important:**
>
> Do not configure subdomains if the parent domain in the same forest is already configured. For more information, see *Multiple-domain support* on page 145.

> ⓘ **Note:** Click the name of an existing domain to edit it. Use the **Delete** and **Undelete** links to remove a domain or cancel a removal request.

Multiple-domain support

If your network uses multiple domains in a single server forest, you can configure one domain within PingFederate Bridge if there is a trust relationship with the other domains you want to use.

This configuration requires a trust relationship among domains, which is established by default when subdomains or separate domains are created within the same forest. For more information, see *https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc773178(v=ws.10)?redirectedfrom=MSDN*.

> ⓘ **Note:**
>
> If you are configuring only one domain, then you also need to configure only one Service Principal Name. For more information, see *Configuring the Active Directory environment* on page 144.

If your network topology consists of multiple forests without a trust relationship between them, you must configure multiple adapter or token processor instances. Map each instance to a separate domain and then map these adapter or token processor instances to your service provider (SP) connections that authenticate using the integrated Kerberos Adapter, the integrated Kerberos Token Processor, or the separately available IWA Adapter.

For information about configuring the PingFederate Bridge Integrated Windows Authentication (IWA) adapter for multiple-domain Active Directory trusts, see *https://support.pingidentity.com/s/article/How-to-configure-IWA-with-multiple-Active-Directory-trusts*.

Adding a domain

You can configure Active Directory domains or Kerberos realms that PingFederate Bridge uses to contact the domain controllers or the key distribution centers (KDCs) for verifying user authentication.

Steps

In the **Manage Domain/Realm** window, enter the required information based on the following table.

Field	Description
Domain/Realm Name	The fully-qualified domain or realm name. For example, companydomain.com
Domain/Realm Username	The ID for the domain or realm account name.
Domain/Realm Password	The password for the domain or realm account.
Domain Controller/Key Distribution Center Host Names (optional)	Specify the host name or IP address of your domain controller or KDC, such as dc01-yvr, and then click **Add**. Repeat this step to add multiple servers.
	If a host name is used, PingFederate Bridge appends the domain to the host name to formulate the fully qualified domain name (FQDN) of the server unless the **Suppress DC / Domain Concatenation** check box is selected.
	If unspecified, PingFederate Bridge uses a DNS lookup.
Suppress DC / Domain Concatenation	Select this check box to specify the desired FQDNs under **Domain Controller/ Key Distribution Center Host Names**. When selected, PingFederate Bridge does not append the domain to the host names.
Test Domain/Realm Connectivity	Tests access to the domain controller or KDC from the administrative-console server.
	When a connection to any of the configured controllers or KDCs is successful, the message Test Successful appears. Otherwise, the test returns error messages near the top of the window.

> ⓘ **Tip:**
>
> For help resolving connectivity issues, select the **Debug Log Output** check box on the **Manage Domain/Realm Settings** window, run the test again, and review the debug messages in the PingFederate Bridge server log.

This test stops at the first successful result when multiple domain controllers or KDCs are specified, so not all servers are necessarily verified. Depending on the network architecture, the engine nodes deployed in a cluster might establish connections differently. As a result, the engine nodes and the console node might connect to different domain controllers or KDCs.

Server

Under **Server**, you can configure protocol settings and administrative accounts, manage licenses, and import and export configurations.

See the following sections:

▪ *Configuration archive* on page 159

Protocol settings

On the **Protocol Settings** window, you can configure the base URL of your PingFederate Bridge environment, and settings for WS-Trust STS authentication, outbound provisioning, and IdP discovery.

Specifying federation information
Federation information identifies your federation deployment to your partners, according to the protocols you support.

About this task

You must provide an ID that uniquely identifies your federation gateway for each protocol you support. For WS-Trust security token service (STS), IDs are required for both SAML 2.0 and SAML 1.x, regardless of browser-based single sign-on (SSO) protocol support or the type of token expected to be issued, to ensure that the STS will perform correctly under all conditions.

> ⓘ **Note:**
>
> Each ID normally applies across all connection partners for a given protocol. However, if your implementation requires different IDs for the same protocol, you can use virtual server IDs. For more information, see *Federation planning checklist*.

Steps

1. Go to **System**# **Server** to open the **Protocol Settings** window.
2. On the **Federation Info** tab, provide the required information.

   For more information, see the following table.

Field	Description
Base URL	The fully qualified host name, port, and path (if applicable) on which the PingFederate Bridge server runs. This field is used to populate configuration settings in metadata files. For more information, see *Metadata export*.
SAML 2.0 Entity ID	This ID defines your organization as the entity operating the server for SAML 2.0 transactions. It is usually defined as an organization's URL or a DNS address, for example: `pingidentity.com`. The SAML SourceID used for artifact resolution is derived from this ID using SHA1.
SAML 1.x Issuer/ Audience	This ID identifies your federation server for SAML 1.x transactions. As with SAML 2.0, it is usually defined as an organization's URL or a DNS address. The SourceID used for artifact resolution is derived from this ID using SHA1.
SAML 1.x Source ID	(Optional) If supplied, the Source ID value entered here is used for SAML 1.x, instead of being derived from the SAML 1.x Issuer/Audience.
WS-Federation Realm	The URI of the realm associated with the PingFederate Bridge server. A realm represents a single unit of security administration or trust.

3. Click **Next** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

Configuring WS-Trust settings
You can configure PingFederate Bridge to require that client applications provide credentials to access the security token service (STS).

About this task

While this is an optional configuration, it is recommended for identity provider (IdP) configurations using the Username Token Processor. For other token processors and token generators, trust in the identity of the client is conveyed within the token itself and verified as part of processing. However, you can still configure authentication requirements to add another layer of security by limiting access to only authenticated clients.

ⓘ **Note:**

You can configure STS authentication to either apply globally to all token formats and for all IdP and service provider (SP) partner connections, or token-to-token mappings, using more fine-grained controls at the connection level through issuance criteria.

Steps

1. Go to **System**# **Server**to open the **Protocol Settings** window
2. On the **WS-Trust STS Settings** tab, click **Configure WS-Trust STS Authentication**.

    Follow the configuration wizard to complete the task. For more information, see *Configuring STS authentication* on page 148
3. Click **Next** and continue with the rest of the configuration.

ⓘ **Tip:**

When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

Configuring STS authentication
You can configure PingFederate Bridge to require that client applications provide credentials to access the STS.

About this task
Although it is an optional configuration, configuring security token service (STS) authentication is recommended for identity provider (IdP) configurations that use the Username Token Processor. For other token processors and token generators, trust in the identity of the client is conveyed within the token itself and verified as part of processing. You can still configure authentication requirements to add another layer of security by limiting access to only authenticated clients.

ⓘ **Note:**

You can configure STS authentication to either apply globally to all token formats and for all IdP and service provider (SP) partner connections, or token-to-token mappings, using more fine grained controls, at the connection level through issuance criteria.

Steps

1. Go to **System**# **Server**# **Protocol Settings**.
2. On the **WS-Trust STS Settings** tab, click **Configure WS-Trust STS Authentication** to open the **WS-Trust STS Settings** window.

3. On the **Authentication Methods** tab, select the **Require HTTP Basic Authentication** check box, the **Require Mutual SSL/TLS Authentication** check box, or both.

If both the **Require HTTP Basic Authentication** check box and the **Require Mutual SSL/TLS Authentication** check box are selected, all clients must provide credentials for both mechanisms.

ⓘ **Important:**

If you select the **Require Mutual SSL/TLS Authentication** check box, you must configure a secondary PingFederate Bridge HTTPS port pf.secondary.https.port in the `run.properties` file. For more information, see *Configuring PingFederate properties*.

4. If you select the **Require HTTP Basic Authentication** check box, manage user accounts on the **HTTP Basic Authentication** tab.

   a. Click **Create User**.
   b. In the **HTTP Basic Authentication**, enter a user name in the **username** field and a password in the **password** field.. Repeat to create additional user accounts for your client applications.
   c. Click **Done**.

ⓘ **Note:**

On the **HTTP Basic Authentication** tab, you can also delete user accounts and update their passwords.

5. If you select the **Require Mutual SSL/TLS Authentication** check box, on the **Mutual SSL Authentication** tab, click **Configure Mutual SSL Authentication**.

   a. On the **Authentication Options** tab, you can select the **Restrict Access by Subject DN** check box and the **Restrict Access by Issuer Certificate** check box. Click **Next**.

   If both options are selected, the client certificate used for authentication to the STS endpoints must meet both sets of restrictions.

   b. If you selected the **Restrict Access by Subject DN** check box, enter one or more subject DNs on the **Allowed Subject DNs** tab.

ⓘ **Note:**

On the **Allowed Subject DNs** tab, you can edit or delete existing entries but you must keep at least one subject DN.

   c. Click **Next**.. When finished, click **Save**.
   d. If you selected the **Restrict Access by Issuer Certificate** check box, on the **Allowed Issuer Certificates** tab, from the **Issuer Certificate** list, select one or more client certificates.
   e. Click **Add**.

   If you have not yet imported the client certificate, click **Manage Certificates** to do so.

ⓘ **Note:**

On the **Allowed Issuer Certificates** tab, you can remove existing entries but you must keep at least one issuer.

   f. On the **Summary** tab, review your mutual SSL/TLS authentication settings. Click **Done**.
      This will take you back to the **WS-Trust STS Settings** window.

6. When you finish configuring WS-Trust STS settings, on the **Summary** tab, review the configuration. To keep your changes, click **Save**.

Configuring outbound provisioning settings

You can select the database that PingFederate Bridge should use internally to facilitate provisioning for service providers when PingFederate Bridge is configured as an identity provider (IdP).

Before you begin

Before configuring outbound provisioning settings, you must enable outbound provisioning through the `pf.provisioner.mode` property in the `<pf_install>`/pingfederate/bin/run.properties file. For more information, see *Configuring PingFederate properties*.

If you want to use failover provisioning, configure the `provisioner.node.id` and `provisioner.failover.grace.period` properties, which are also located in `<pf_install>`/pingfederate/bin/run.properties. These properties are described in *Deploying provisioning failover*.

About this task

The database stores the state of synchronization between the source datastore and the target datastore, enabling periodic checking to determine whether updates are required at the target site. PingFederate Bridge checks the source datastore for changes every minute by default. As needed, you can change the provisioning synchronization frequency on this tab as well.

> ⓘ **CAUTION**:
>
> Use the built-in HSQLDB only for trial or training environments. For testing and production environments, always use a secured external storage solution for proper functioning in a clustered environment.
>
> Testing involving HSQLDB is not a valid test. In both testing and production, it might cause various problems due to its limitations and HSQLDB involved cases are not supported by PingIdentity.

> ⓘ **Note:**
>
> PingFederate Bridge is tested with Amazon Aurora (MySQL and PostgreSQL), Microsoft SQL Server, Oracle Database, Oracle MySQL, and PostgreSQL as internal provisioning datastores. A demonstration-only, embedded HSQLDB database is installed by default. Scripts to aid setup are in the directory `<pf_install>/pingfederate/server/default/conf/provisioner/sql-scripts`.

> ⓘ **CAUTION**:
>
> A pre-installed, default HSQLDB database is selected for initial setup and testing. However, we strongly recommend that you use the built-in HSQLDB only for trial or training environments. For testing and production environments, always use a secured external storage solution for proper functioning in a clustered environment.
>
> Testing involving HSQLDB is not a valid test. In both testing and production, it may cause various problems due to its limitations and HSQLDB involved cases are not supported by PingIdentity.

Steps

1. Go to **System**# **Server**# **Protocol Settings**.
2. On the **Outbound Provisioning** tab, from the **Provisioning Data Store** list, select a datastore.

   If the datastore you want is not shown in the list, PingFederate Bridge is not yet configured to access the store. Click **Manage Data Stores** to create a connection to the datastore.

3. Change the **Synchronization Frequency** value.

   The default value is `60` seconds.

4. Click **Next** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

Configuring standard IdP Discovery

SAML 2.0 identity provider (IdP) Discovery provides a cookie-based look-up mechanism to identify a user's IdP dynamically during a service provider (SP)-initiated single sign-on (SSO) event when the IdP is not otherwise specified.

About this task

This mechanism can be helpful in cases where an SP might be a hub for several IdPs in an identity federation.

In addition to supporting SAML 2.0 IdP Discovery, PingFederate Bridge provides a cross-protocol, proprietary mechanism allowing a PingFederate Bridge SP server to write a persistent browser cookie. The cookie contains a reference to the IdP partner with whom the user previously authenticated for SSO. For more information, see *Configuring IdP discovery using a persistent cookie*.

> ⓘ **Tip:**
>
> An SP can also include the discovery mechanism within the application. For instance, an SP can provide vanity URLs to isolate one set of end users from the others based on the URL of the requested resources. Another possible solution is to provide a user interface for the end users to enter information about their identity providers. With this approach, the application can start an SP-initiated SSO request with information about the IdP.

In the standard scenario, when a user requests access to a protected resource on the SP, common-domain browser cookies are used to determine where a user has authenticated in the past. Using this information, a PingFederate Bridge server can determine which IdP connection to use for sending an authentication request.

As an IdP Discovery provider, PingFederate Bridge can serve in up to three different roles: common domain server, common domain cookie writer, and common domain cookie reader. Each of these roles is necessary to support IdP Discovery. The roles can be distributed across multiple servers at different sites.

Common domain server

> In this role the PingFederate Bridge server hosts a domain that its federation partners share in common. The common domain server allows partners to manipulate browser cookies that exist within that common domain. PingFederate Bridge can serve in this role exclusively or as part of either an IdP or an SP federation role, or both.

Common domain cookie writer

> When PingFederate Bridge is acting in an IdP role and authenticates a user, it can write an entry in the common domain cookie, including its federation entity ID. An SP can look up this information on the common domain, not the same location as the common domain server described above.

Common domain cookie reader

> When PingFederate Bridge is acting as an SP and needs to determine the IdPs with whom the user has authenticated in the past, it reads the common domain cookie. Based on the information

contained in the cookie, PingFederate Bridge can then initiate an SSO authentication request using the correct IdP connection.

Steps

1. Go to the **Protocol Settings** window.
2. On the **IdP Discovery** tab, click **Configure IdP Discovery**.
3. On the **Domain Cookie Settings** tab, choose the discovery role or roles of your PingFederate Bridge server.
4. On the **Common Domain Service** tab, configure as follows.

Field	Description
Base URL of the PingFederate Common Domain Service	Enter the base URL of the PingFederate Bridge common domain service.
	A common domain service is where PingFederate Bridge reads or writes authentication information contained in shared cookies, as determined by whether your site is an SP or IdP, respectively. The service is shared if your PingFederate Bridge server is acting in both roles. You must use HTTPS for the common domain.
Pass Phrase and **Confirm Pass**	Enter and confirm the pass phrase that web applications must use to access the domain.

5. On the **Local Common Domain Server** tab, configure the required settings.

   A local common domain server is where PingFederate Bridge reads (as an SP) or writes (as an IdP) a common domain cookie (CDC) for IdP Discovery.

Field	Description
Common Domain	Enter the common domain.
	Your entry must include an initial period (.), as in the following example.
	`.example.com`
Cookie Lifetime (Days)	Enter the lifetime of the CDC in days. The range is `1` to `1825` days. To indicate a non-persistent session cookie, enter `-1`.
Pass Phrase and **Confirm Pass**	Enter and confirm the pass phrase that web applications must use to access the domain.

6. On the **Summary** tab, review and modify settings as needed. Then click **Save**.

   The administrative console brings you back to the **IdP Discovery** tab.
7. Click **Next** and continue with the rest of the configuration.

   > ⓘ **Tip:**
   >
   > When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

8. Perform one of the following actions to enable the setting of the common domain cookie at runtime:

   - Make sure that, prior to launching any SSO events, the web application that implements IdP Discovery sets the cookie using the `/idp/writecdc.ping` application endpoint intended for that purpose.
   - Enable setting the cookie at runtime during SSO events by selecting the **IdP Discovery** check box on the **Connection Options** tab for the desired SP connection.

IdP protocol endpoints

PingFederate Bridge provides a list of identity provider (IdP) protocol endpoints and exportable metadata for your configuration.

You can find a list of applicable SAML, WS-Federation, and WS-Trust STS endpoints in **System**# **Endpoints**# **IdP Endpoints**. The pop-up window displays only those endpoints related to the federation protocols enabled on **System**# **Server**# **Protocol Settings**# **Federation Info**. These endpoints are built into PingFederate Bridge and cannot be changed.

Your federation partners or security token service (STS) clients need to know the applicable IdP services endpoints to communicate with your PingFederate Bridge server. Configured service endpoints for SAML connections are included in metadata export files.

PingFederate Bridge provides a favicon for all protocol endpoints. For more information, see *Customizing the favicon for application and protocol endpoints*.

The following table describes each endpoint.

Service	URL and Description
Single Logout Service (SAML 2.0)	`/idp/SLO.saml2` The URL that receives and processes logout requests and responses.
Single Sign-on Service (SAML 2.0)	`/idp/SSO.saml2` The SAML 2.0 implementation URL that receives authentication requests for processing.
Artifact Resolution Service (SAML 2.0)	`/idp/ARS.ssaml2` The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel. See the note at the end of this table.
Attribute Query Service (SAML 2.0)	`/idp/attrsvc.ssaml2` The SAML implementation that receives and processes attribute requests. See the note at the end of this table.
Single Sign-on Service (SAML 1.x)	`/idp/isx.saml1` The SAML 1.x implementation of IdP intersite transfer service (ISX) to which clients are redirected for single sign-on (SSO) requests.
Artifact Resolution Service (SAML 1.x)	`/idp/soap.ssaml1` The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel. See the note at the end of this table.
Single Sign-on Service (WS-Federation)	`/idp/prp.wsf` The WS-Federation implementation URL that receives and processes security-token requests and single log-out (SLO) messages.

Service	URL and Description

WS-Trust STS (two endpoints)	`/idp/sts.wst`
	
	The SOAP endpoint that receives and processes security-token requests from STS clients (web service clients at the IdP site) to be exchanged for a SAML token based on the configured service provider (SP) connection.
	
	`/pf/sts.wst`
	
	Initiates direct STS token-to-token exchange and token validation from an IdP token processor to an SP token generator, when that feature is configured. For more information, see *Token translator mappings*.
	
	ⓘ **Note:**
	If multiple token-processor instances of the same type are configured for the same connection or token-to-token mapping, a query parameter, TokenProcessorId, must be added to either of these endpoints. For more information, see *Managing token processors*.
	
	See the note at the end of this table.

> ⓘ **Important:**
> If mutual SSL/TLS is used for authentication, a secondary PingFederate Bridge listening port must be configured and used by partners or STS clients for the relevant endpoints—`*.ssaml*` and `*.wst` For more information, see *Configuring PingFederate properties*.

Virtual server ID support

For SAML connections using multiple virtual server IDs, each virtual server ID has its own set of protocol endpoints. For more information, see *Multiple virtual server IDs*. You can export connection metadata for your partner from **System**# **Protocol Metadata**# **Metadata Export**. For more information, see *Exporting connection-specific SAML metadata*.

For WS-Federation (and SAML) connections using multiple virtual server IDs, you can provide your partner the federation metadata endpoint, `/pf/federation_metadata.ping`, with the PartnerSpId and vsid parameters, as in the following example.

Partner's entity ID	Your virtual server ID	Federation metadata URL
SP	idev1	https://www.example.com/pf/federation_metadata.ping?PartnerSpId=SP&vsid=idev1
	idev2	https://www.example.com/pf/federation_metadata.ping?PartnerSpId=SP&vsid=idev2

In this example, the base URL and the runtime port of your PingFederate Bridge server are www.example.com and 443, respectively.

When the request includes the vsid parameter, the federation metadata endpoint returns information that is specific for a given virtual server ID.

For WS-Trust STS, you can provide your partner the STS metadata endpoint `/pf/sts_mex.ping` with the PartnerSpId and vsid parameters. When the STS metadata request includes the vsid parameter, the STS metadata endpoint returns information that is specific for a given virtual server ID.

For more information about these metadata endpoints, see *System-services endpoints*.

> ⓘ **Note:** The virtual server ID concept does not apply to the `/pf/sts.wst` endpoint because token-to-token exchange does not involves any connections. As needed, you can pass the token-to-token endpoint to your partners as-is.

Reviewing protocol settings

On the **Summary** tab, you can review and save your protocol settings.

Steps

- To amend your configuration, click the corresponding tab title, then follow the configuration wizard to complete the task.
- To keep your changes, click **Done** and continue with the rest of the configuration.

> ⓘ **Tip:**
>
> When editing an existing configuration, you can also click **Save** as soon as the administrative console offers the opportunity to do so.

- To discard your changes, click **Cancel**.

Administrative accounts

PingFederate Bridge supports five authentication schemes.

The authentication schemes are:

- Native authentication
- LDAP authentication
- RADIUS authentication
- Certificate-based authentication
- OIDC-based authentication

For role-based access control, PingFederate Bridge provides two account types and four administrative roles, as shown in the following table.

PingFederate Bridge User Access Control

Account type	Administrative role	Access privileges
Admin	User Admin	Create users, deactivate users, change or reset passwords, and install replacement license keys.
Admin	Admin	Configure partner connections and most system settings, except the management of local accounts and the handling of local keys and certificates.

Account type	Administrative role	Access privileges
Admin	Expression Admin	Map user attributes by using the expression language, Object-Graph Navigation Language (OGNL).

> ⓘ **Important:**
>
> Only Administrative users who have both the Admin role and the Expression Admin role:
>
> - Can be granted the User Admin role. This restriction prevents non-Expression Admin users from granting themselves the Expression Admin Role.
> - Can be granted write access to the file system or directory where PingFederate Bridge is installed. This restriction prevents a non-Expression Admin user from placing a `data.zip` file containing expressions into the `<pf_install>/pingfederate/server/default/deploy` directory, which would introduce expressions into PingFederate Bridge.

Account type	Administrative role	Access privileges
Admin	Crypto Admin	Manage local keys and certificates.
Auditor	Not applicable	View-only permissions for all administrative functions. When the **Auditor** role is assigned, no other administrative roles can be set.

> ⓘ **Note:**
>
> All four administrative roles are required to access and make changes through the following services:
>
> - The `/bulk`, `/configArchive`, and `/configStore` administrative API endpoints
> - The **Configuration Archive** window, accessed from **System**# **Server**, in the administrative console
> - The **Connection Management** configuration item on the **Service Authentication** window, accessed from **Security**# **System Integration**

For native authentication, access and authorization are controlled by the local accounts defined on the **Administrative Accounts** window.

As needed, you can switch from native authentication to an alternative console authentication. Access and authorization are defined in the respective configuration file.

An administrative user can sign on from more than one browser or location. Moreover, multiple administrative users can sign on to the PingFederate Bridge administrative console at a time. You can optionally restrict the administrative console to one administrative user at a time by modifying the pf.console.login.mode property in the `<pf_install>/pingfederate/bin/run.properties` file. Regardless of the property configuration, any number of auditors can sign on at any time.

> ⓘ **Note:**
>
> For security, after three failed sign-on attempts from the same location within a short time period, the administrative console and the administrative API will temporarily lock out further attempts by the same user. The user must wait one minute to try again.

Local accounts defined on the **Administrative Accounts** window are shared between the administrative console and the administrative API if they are both configured to use native authentication, the default. If the administrative console is configured to use an alternative console authentication, the **Administrative Accounts** window appears only if the administrative API is left to use native authentication, and vice versa.

> ⓘ **Tip:**
>
> If you have connected PingFederate Bridge to PingOne for Enterprise, you can also single sign-on from the PingOne admin portal to the administrative console.

Managing local accounts and role assignments

You can create, modify, update, or deactivate accounts in the **Administrative Accounts** window.

Steps

1. Go to **System# Server# Administrative Accounts**, and then perform any of the following actions.

Task	Steps
Create a local account	**a.** On the **Administrative Accounts** window, click **Create User**. **b.** On the **User Information** tab, enter a username and other optional information. > ⓘ **Note:** > If you want PingFederate Bridge to notify the user about password changes via email, you must supply an email address. **c.** On the **Password Generation** tab, enter a password or click **Generate one-time password** to generate a random password for the account. > ⓘ **Note:** > Upon successful authentication, the user will be required to change the password of the account immediately. **d.** On the **Summary** tab, review your configuration, modify as needed, and then click **Done**. **e.** On the **Administrative Accounts** window, select the applicable account type, **Auditor** or **Admin**, and one or more administrative roles for an **Admin** account. **f.** Repeat these steps to create additional accounts.
Modify user information	**a.** On the **Administrative Accounts** window, select the account by its username. > ⓘ **Note:** > Applicable only to active accounts. **b.** On the **User Information** window, update the record, and then click **Done**. **c.** Repeat these steps to update other accounts.

Task	Steps
Update role assignments	a. Select a different account type, **Auditor** or **Admin**, for one or more accounts. b. Select or clear the check boxes that correspond to the three administrative roles, **User Admin**, **Admin**, and **Crypto Admin** for one or more accounts. ⓘ **Note:** Applicable only to the **Admin** accounts.
Deactivate or reactive a native	a. Click **Deactivate** or **Activate** under **Action**. b. Repeat this step to deactivate or reactive other accounts. ⓘ **Note:** For traceability and accountability purposes, local accounts cannot be deleted Their records are retained and they can be reactivated if needed.

2. To keep your configuration, click **Save**.

Setting or resetting passwords

On the **Password Generation** tab, you can generate temporary passwords as you create local accounts for new users.

About this task

You can also assign temporary passwords for existing users who forget their passwords. Upon successful authentication, the users are required to change their passwords immediately.

Steps

1. Go to **System**# **Server**# **Administrative Accounts**.
2. To generate a random password for the account, on the **Password Generation** tab, enter a password or click **Generate one-time password**, and then click **Next**.

License management

PingFederate licensing is handled differently depending on whether you are installing and setting up PingFederate for the first time, or upgrading your existing PingFederate installation to a later version.

Initial PingFederate installations

During the initial setup for a new PingFederate installation:

- If you choose to connect to PingOne for Enterprise, PingFederate obtains and installs an evaluation PingFederate Bridge license from PingOne.
- If you choose not to connect to PingOne for Enterprise, you are prompted to upload a license file.

Depending on your licensing agreement, your license might have an expiration date. If your license key is going to expire, or has expired recently, you can import a new license file to replace the existing license key through the administrative console.

The administrative console displays a warning message ahead of the expiry of your license. Optionally, you can configure PingFederate to notify the administrators ahead of the license expiration date.

PingFederate upgrades

If you choose to upgrade PingFederate using the Upgrade Utility, your current PingFederate license is automatically copied to the target installation if it is valid. If your current license is not valid, you must obtain a new license and specify its full path and filename when performing the upgrade.

If you choose not to use the Upgrade Utility, you must specify the full path and filename of a valid license when performing the upgrade.

Reviewing license information

You can review a summary of your PingFederate license in the **License** window.

About this task

ⓘ **Important:** If the license specifies an expiration date, the license expires at the beginning of that day.

Steps

1. On the Help menu, click **About**.
2. In the dialog box, click **View / Update License**. The **License** window opens.

   ⓘ **Tip:** Another way to open the **License** window is to go to **System**# **Server**# **License**.

Configuration archive

You can use configuration archives as backup files for the current PingFederate Bridge installation.

PingFederate Bridge automatically creates a time-stamped configuration (`.zip`) archive every time an administrator signs on to the administrative console and before an existing archive is imported. The archives are stored in the *`<pf_install>`*`/pingfederate/server/default/data/archive` directory.

The automatic backup process typically completes without delays. For deployments with hundreds of connections or OAuth clients, or both, administrators can configure PingFederate Bridge to create configuration archives periodically instead.

Additionally, administrators can export the current configuration to a `.zip` file in the **Configuration Archive** window. This window is only available to administrators whose accounts have been assigned the User Admin, Admin, Crypto Admin, and Expression Admin roles.

ⓘ **Note:**

The Expression Admin role must be assigned to give administrators sufficient permissions to create configuration archives.

ⓘ **Warning:**

The backup file contains your complete configuration. To protect your data, confirm the backup file is protected with appropriate security controls in place before exporting it.

Sharing the archive is a security risk because the private keys are stored in the archive. An archive should only be shared if the security of that instance is not important, such as a development or test environment.

On the **Configuration Archive** window, administrators can import an existing archive for immediate deployment into a running PingFederate Bridge server.

Administrators can also deploy a configuration archive manually by copying the `.zip` file to the `<pf_install>/pingfederate/server/default/data/drop-in-deployer` directory. After copying the `.zip` file, it must be renamed to `data.zip`.

Configuration archives are intended for administrative-console configuration only. The following files are not included in the archives:

- Launch scripts in the `<pf_install>/pingfederate/bin` and `<pf_install>/pingfederate/sbin` directories.
- Web container configuration files in the `<pf_install>/pingfederate/etc` directory.
- Log files in the `<pf_install>/pingfederate/log` directory.
- Database drivers and program files from adapters and any other plugins in the `<pf_install>/pingfederate/server/default/lib` and `<pf_install>/pingfederate/server/default/deploy` directories.
- Other files, including the license file, the advanced cluster configuration files, and the user-facing email and HTML templates, in the `<pf_install>/pingfederate/server/default/conf` directory.

If any changes have been made to files that are not part of the configuration archive, those files must be preserved manually.

> ⓘ **Tip:**
>
> You can export a configuration archive, extract the `.zip` file, and determine whether specific files are part of the configuration archive, or not.

> ⓘ **Important:**
>
> Draft connections in archives are not imported. Complete any unfinished partner connections if you want to include them in a full backup archive or in an archive to be used for configuration migration.

Exporting an archive
On the **Configuration Archive** window, you can export the current administrative-console configuration to a `.zip` file. This window is only available to administrators whose accounts have been assigned the User Admin, Admin, Expression Admin, and Crypto Admin roles.

Steps

- On the **Export** tab, click **Export**, then save the `.zip` file.

> ⓘ **Warning:**
>
> The backup file contains your complete configuration. To protect your data, confirm the backup file is protected with appropriate security controls in place before exporting it.
>
> Sharing the archive is a security risk because the private keys are stored in the archive. An archive should only be shared if the security of that instance is not important, such as a development or test environment.

Importing an archive
On the **Configuration Archive** window, you can import an administrative-console configuration from a `.zip` file. This window is only available to administrators whose accounts have been assigned the User Admin, Admin, and Crypto Admin roles.

About this task

When an administrator initiates deployment of a configuration archive using the **Import** tab, PingFederate Bridge displays error messages if there are any missing plugin components, such as adapters, database drivers, or token translators, on which the archive depends, or any mismatches of PingFederate Bridge licensing authorization. The administrator can choose to force the deployment and then install the necessary files later.

ⓘ **Note:**

Installation of any missing database drivers or other third-party libraries will require a restart of PingFederate Bridge.

ⓘ **CAUTION:**

Deploying a configuration archive, either manually or by using the administrative console, always overwrites all existing configuration data.

Steps

1. On the **Import** tab, choose the desired configuration archive from your system.
2. Select the **Force Import** check box if you want PingFederate Bridge to deploy the archive regardless of whether dependency errors are detected.

   ⓘ **Important:**

   If you make this selection, consult the server start-up console or the server log for any messages concerning missing plugin components or other errors.

3. Click **Import**.

   The administrative console prompts you to confirm the import process.

External systems

Many use cases require communications between PingFederate Bridge and other systems, such as PingOne or a database server. **System# External Systems** is where you can set up and maintain such integrations.

Connections to PingOne

Some PingFederate plugins that access services provided by PingOne can leverage a secure connection between PingFederate and PingOne.

For example, when configuring instances of PingFederate's PingOne MFA and PingOne Risk plugins, you can select a connection and one of the connection's PingOne environments. You do not need to enter credentials and other connection details for each instance. Also, you do not need to update each plugin instance individually when, for example, you replace the connection's credential.

The central element of a connection is its credential, which an administrator generates on PingOne and copies to PingFederate when they create the connection. During runtime, a PingFederate plugin

instance that was configured to use the connection gets the credential and uses it to access a PingOne environment.

Administrators with access to both PingFederate and PingOne create the connection. For more information, see *Creating connections to PingOne* on page 162.

You can create multiple connections but most deployments require only one enabled connection. Typically, you use only one connection from a PingFederate cluster to a given PingOne organization. A single connection can be configured to provide access to multiple environments in PingOne.

Cards on PingFederate's **System**# **External Systems**# **PingOne Connections** window display information about each connection to PingOne. When the card is collapsed, it shows only the connection's name, ID, and status toggle switch. Use the toggle switch to enable and disable the connection on PingFederate.

> ⓘ **Note:** Administrators can disable a connection on PingFederate or PingOne. Plugins can use a connection only if it is enabled on both PingFederate *and* PingOne.

When a connection card is expanded, it shows more information about the connection on tabs, including the status of the main connection and the plugin connections, if any.

Tab	Description
Summary	In the **Summary** tab's **Summary Details** section, the **Description** and **Created** time come from PingFederate. The **Connection ID in PingOne**, **Credential ID**, **Organization Name**, and **Region** come from the connection credential.
	In the **Summary** tab's **Configuration** section, the image shows:
	▪ the status of the connection between PingFederate and PingOne
	▪ which plugin types can use this kind of connection and are installed on PingFederate, if any
	▪ which service on PingOne each of those plugin types is designed to use
	▪ whether any instances of those plugin types are configured to use this particular connection
Usage	The **Usage** tab lists each plugin instance on PingFederate that is configured to use this connection. The instances are grouped by plugin category.
Environments	The **Environments** tab shows which environments on PingOne the connection is able to access. You assign environments to connections in PingOne. In PingFederate, when you configure a plugin to use a connection, you select the connection and one of its environments.

Creating connections to PingOne

Administrators can create connections between PingFederate and PingOne that some PingFederate plugins can use to access services provided by PingOne.

About this task

In the following procedure, you create a connection profile and credential in PingOne. Then you use the credential in PingFederate to establish the connection. So you need administrative access to both applications.

> ⓘ **Note:** You cannot use the same credential in multiple extant connections.

After creating the connection, you can use it to configure PingFederate plugins that support connections to PingOne, such as the PingOne MFA and PingOne Risk plugins.

For more information about these connections, see *Connections to PingOne* on page 161 in the PingFederate documentation and *PingFederate connections* in the PingOne documentation.

Steps

1. In PingOne, create a connection profile and credential:

    a. Go to **Connections**# **Product Platform**# **PingFederate**.

    b. Click **+ Add Connection**.

    c. On the **Create Connection Profile** window, enter a **Connection Name** and **Description**.

    d. Click **Save and Continue**.

    e. On the **Establish Connection with PingFederate** window, copy the connection credential by clicking **Copy to Clipboard**.

    f. If you will not perform step 2 now, paste and save the credential in a text file because this is the only time you can view and copy the credential.

2. In PingFederate, establish the connection:

    a. Go to **System**# **External Systems**# **PingOne Connections**.

    b. On the **PingOne Connections** window, click **+ Add Connection**.

    c. On the **Add Connection** window, paste the credential that was created in step 1.

    d. Enter a new **Connection Name** or keep the auto-populated name, which is the name of your organization on PingOne. Ensure the name is unique.

    e. Enter a **Connection Description**.

    f. Click **Save**. The **PingOne Connections** window shows the new connection. The connection is enabled by default.

Next steps

After you create a connection, you can:

- use the connection to configure PingFederate plugins that support these kinds of connections
- change which environments in PingOne the connection can access
- replace or revoke the connection's credentials
- disable the connection
- change the connection's name or description

For more information, see *Modifying connections to PingOne* on page 163.

Modifying connections to PingOne

Administrators can modify a connection between PingFederate and PingOne by, for example, adding a PingOne environment to it, replacing its credential, changing its description, or disabling it

> ⓘ **Note:** If you have multiple connections to PingOne, to avoid confusion, you can identify a connection by its connection ID in both PingFederate and PingOne. The connection ID never changes, whereas its name can be changed.

Editing connection names and descriptions

You can change the names and descriptions of connections between PingFederate and PingOne.

About this task

The following procedure describes how to change the name and description of the connection in PingFederate. You can also change them in PingOne.

> ⓘ **Note:** Changing the name and description in PingFederate does not change them in PingOne, and vice versa. You cannot change a connection's ID.

Steps

1. In PingFederate, go to **System**# **External Systems**# **PingOne Connections**.
2. On the **PingOne Connections** window, open the connection's card by clicking its expand icon.
3. Click the pencil icon.
4. Change the **Connection Name**, **Connection Description**, or both.
5. Click **Save**.

Disabling and enabling connections

You can disable a connection between PingFederate and PingOne in either of those applications; however, to enable a connection, it must be enabled in both applications.

Steps

1. To disable or enable a connection in PingFederate:
   a. Go to **System**# **External Systems**# **PingOne Connections**.
   b. On the connection's card, click the toggle switch to change the connection's state.
2. To disable or enable a connection in PingOne:
   a. Go to **Connections**# **Product Platform**# **PingFederate**.
   b. On the connection's card, click the toggle switch to change the connection's state.

Replacing connection credentials

You can replace or revoke the credential of a connection between PingFederate and PingOne.

About this task

The following procedure describes how to replace a connection's credential, which requires administrative access to both applications.

Steps

1. In PingOne:
   a. Go to **Connections**# **Product Platform**# **PingFederate**.
   b. On the **PingFederate Connections** window, open the connection's card and go to the **Configuration** tab.
   c. Click the pencil icon.
   d. Click **+ Add**.
      PingOne generates a new credential, adds it to the tab, and opens a dialog box.
   e. In the dialog box, click the credential to save it to the clipboard.
   f. Close the dialog box.
   g. If you will not perform step 2 now, paste and save the credential in a text file because this is the only time you can view and copy the credential.
2. In PingFederate:
   a. Go to **System**# **External Systems**# **PingOne Connections**.
   b. On the **PingOne Connections** window, open the connection's card.
   c. Click the pencil icon.
   d. Paste the new credential that you created in step 1 in the **Credential** field.
   e. Click **Save**.

Next steps

After you replace a connection's credential, you can go to PingOne and revoke the old credential.

Modifying which environments connections can access
PingOne administrators can specify which environments in PingOne a connection can access.

In PingOne, environments are assigned to roles and roles are assigned to connections. When a connection is created, PingOne assigns it default roles and environments. A connection can access all the environments assigned to its roles. For more information, see *Edit a connection: Add roles and responsibilities* in the documentation for PingOne.

Managing PingOne for Enterprise settings

You can configure PingOne for Enterprise's general settings and single sign-on (SSO) settings, and enable and configure a RADIUS server to integrate PingID with a VPN.

About this task

Go to **System**# **External Systems**# **PingOne for Enterprise Settings**, configure various PingOne for Enterprise integration settings and optionally enable and configure a built-in RADIUS server to integrate PingID with your VPN.

Deploying cluster servers

A PingFederate cluster consists of multiple nodes, each of which are likely running on a dedicated host system.

About this task

In a cluster, there are two types of nodes: engine nodes and an administrative console node. Engine nodes service end user traffic, and multiple nodes are recommended to ensure high availability for your deployment. Only one administrative console node can be running in a given cluster. This node hosts the user interface and APIs that you use to configure the cluster. Additionally, the administrative console node manages the following runtime functions:

- Performing periodic configuration archive backup
- Cleaning expired persistent authentication sessions
- Cleaning expired access grants
- Updating connections from metadata URLs, including SP connections if configured, and sends email notifications
- Performing automatic rotation of signing certificates if enabled

ⓘ **Note:**

Additional steps are required to set up failover for provisioning. If you are grouping servers exclusively to provide for provisioning failover, skip these steps and see *Deploy provisioning failover*.

These steps describe how to configure and deploy clustered PingFederate servers by editing each node in the `<pf_install>`/pingfederate/bin/run.properties.

Steps

1. Install PingFederate on each node in a cluster.

2. Edit the clustering properties of each node in the *<pf_install>*/pingfederate/bin/
run.properties file. See the following table for information about each property.

Property	Description

pf.operational.mode	Controls the operational mode of the PingFederate server. PingFederate supports the following modes:
	
	STANDALONE (default)
	
	    This server is a standalone instance that runs both the administrative console and runtime engine.
	
	ⓘ **Important:**
	The value STANDALONE should only be used in a cluster where session-state management is not needed for any reason and configuration-archive deployment is used as the configuration synchronization method.
	
	CLUSTERED_CONSOLE
	
	    This server is part of a cluster and runs only the administration console.
	
	ⓘ **Important:**
	Only one node in a cluster can run the administrative console.
	
	CLUSTERED_ENGINE
	
	    This server is part of a cluster and runs only the runtime engine.

pf.cluster.node.index	Defines a unique index number for the server in a cluster. The index number is used to identify peers and optimize inter-node communication. The allowed range is 0 to 65535. If no value is set for the node index, the system assigns an auto-generated value in the range of 0 to 2147483647. This property has no default value. If you specify an index number, you can configure instances of the Cluster Node Authentication Selector and place them in authentication policies to customize authentication requirements based on the runtime node servicing a request.
pf.cluster.auth.pwd	Sets the password that each node in the cluster must use to authenticate when joining the cluster. This prevents unauthorized nodes from joining a cluster. The value can be any string, or blank. ⓘ **Note:** Consider using a randomly-generated key with 22 or more alphanumeric characters. We recommend that you obfuscate the password. For information about the **obfuscate** command-line utility, see its built-in help. All nodes in a cluster must share the same value, blank or otherwise.

Property	Description
pf.cluster.encrypt	Indicates whether to encrypt network traffic sent between nodes in a cluster. The possible values are `true` or `false` (default).
	When set to `true`, communication within the cluster is encrypted with a symmetric key derived from the value of the pf.cluster.auth.pwd property.
	ⓘ **Important:** When the pf.cluster.encrypt property is set to `true`, you must provide a value for the pf.cluster.auth.pwd property. Otherwise PingFederate aborts during its startup process.
	All nodes in a cluster must have the same value for this property.
pf.cluster.encryption.keysize	The length of the key that PingFederate Bridge takes into consideration when deriving the symmetric key from the value of the pf.cluster.auth.pwd property for the purpose of encrypting network traffic sent between nodes in a cluster. Required only when the pf.cluster.encrypt is set to `true`.
	All nodes in a cluster must have the same value set for this property.
	The default value is `128`.
pf.cluster.bind.address	Defaults to `NON_LOOPBACK`, which leaves the system to choose an available non-loopback IP address. Alternatively, enter an IP address of the network interface to which the cluster communication should bind. For machines with more than one network interface, provide a specific IP address.
	You can use this property to increase performance (particularly with UDP) and improve security by segmenting cluster-communication traffic onto a private network or VLAN.
	ⓘ **Tip:** Besides `NON_LOOPBACK` or an IP address, you can also use other values supported by JGroups. For more information, see the bind_addr special values in *JGroups documentation*.
	ⓘ **Important:** This field does not support DNS name. Use the default value *NON_LOOPBACK* or replace it with an IP address.
pf.cluster.bind.port	Specifies the port associated with the pf.cluster.bind.address property or with the default network interface used.
	This is the port used by other cluster members during their discovery process, usually via the pf.cluster.tcp.discovery.initial.hosts property.
	The default value is `7600`.
pf.cluster.failure.detection.bind.port	Indicates the bind port of a server socket that is opened on the given node and used by other nodes as part of the cluster's failure-detection mechanisms. If set to `0` or unspecified, a random available port is used. The default value is `7700`.

Property	Description
pf.cluster.transport.protocol	Indicates the transport protocol used for cluster communication. Values are `udp` or `tcp`. The default value is *tcp*. All nodes in a cluster must have the same value set for this property.
	Use UDP when IP multicasting is enabled in the network environment and the majority of cluster traffic is point-to-full-group. You must also configure both the pf.cluster.mcast.group.address and pf.cluster.mcast.group.port properties.
	Use TCP for geographically dispersed servers or when multicast is not available or disabled for some other reason. For example, when using routers that do not support multicast messaging. TCP may also be appropriate if your cluster configuration employs more point-to-point or point-to-few messaging than point-to-group.You must also configure the pf.cluster.tcp.discovery.inital.hosts property.
	ⓘ **Note:** This property is a reference to a protocol-stack XML configuration file located in the `<pf_install>/pingfederate/server/default/conf/` directory. Two stacks are provided: one for UDP multicast and one for TCP. You can customize either stack or add to it as needed by modifying the associated configuration file.
pf.cluster.mcast.group.address	Defines the IP address shared among nodes in the same cluster for UDP multicast communication; required when UDP is set as the transport protocol. The valid range is `224.0.0.0` to `239.255.255.255`. Some addresses in this range are reserved for other purposes. This property is not used for TCP.
	All nodes in a cluster must have the same value set for this property.
	The default value is *239.16.96.69*.
pf.cluster.mcast.group.port	Defines the port in conjunction with the pf.cluster.mcast.group.address property value. This property is not used for TCP configurations.
	All nodes in a cluster must have the same value set for this property.
	The default value is `7601`.
pf.cluster.tcp.discovery.initial.hosts	Designates a static list of PingFederate Bridge servers to be contacted for cluster membership information when discovering, joining, and rejoining the cluster. This value is required when TCP is set as the transport protocol. The value is a comma-separated list of host names (or IP addresses) and their cluster bind ports, for example, `host1[7600],10.0.1.4[7600],host7[1033],10.0.9.45[2231]`.
	When using static discovery, add at least one node for the cluster to know in advance. This property should contain all nodes in the cluster (including itself) to increase the likelihood of new members finding and joining the cluster.
	When using dynamic discovery, leave this property blank and enable dynamic discovery in the `<pf_install>/pingfederate/server/default/conf/tcp.xml` file. For more information, see *Enabling dynamic discovery for clustering*.

Property	Description
pf.cluster.adaptive	Indicates whether runtime state-management services should use the adaptive clustering architecture.
	The default value is `true` for new installations and `false` for upgrades.
pf.cluster.diagnostics.enabled	`false` turns off JGroups diagnostics. `true` turns it on.
	The default value is `false`.
pf.cluster.diagnostics.addr and pf.cluster.diagnostics.port	The multicast address and port this node listens on for diagnostic messages.
	The default values are `224.0.75.75` and `7500`, respectively. Do not change the default values.
node.tags	Defines the tags associated with this node.
	Configuration is optional. When configured, PingFederate Bridge considers this property when processing requests. For example, you can use tags to determine the datastore location that this PingFederate node communicates with. You can also use tags in conjunction with authentication selectors and policies to define authentication requirements.
	You can specify one tag.
	`node.tags=north`
	You can also specify a list of prioritized, space-separated tags.
	`node.tags=1 123 234`
	Tags cannot contain spaces.

3. Optional: Edit configuration files in each node that control the cluster protocol and runtime state-management service. For more information, see *Runtime state-management architectures* and *Runtime state-management services*.
4. Optional: If outbound provisioning is configured for your site and you want to provide failover capabilities, identify and configure the provisioning failover nodes. For more information, see *Deploy provisioning failover*.
5. Start or restart PingFederate on all nodes.
6. Sign on to the administrative console.
7. If you have not done so, import your PingFederate Bridge license. For more information, see *License management* on page 158.
8. On the **System**# **Server**# **Cluster Management** window, click **Replicate Configuration** to push the license information from the console node to all engine nodes.

Results

After the clustered environment is set up, you can start configuring PingFederate through the administrative console. When PingFederate detects a change, it prompts you to replicate the configuration to all engine nodes.

Configuring end-user browsers

You can configure browsers at your site in order to use the Kerberos Adapter to authenticate users.

The client-side configuration requires the base URL or an applicable virtual host name of your PingFederate Bridge environment. Base URL is defined on the **System**# **Server**# **Protocol Settings**# **Federation Info** tab. To see a list of defined virtual host names, if configured, go to **System**# **Server**# **Virtual Host Names**.

> ⓘ **Important**:
>
> If the browser is not properly configured, the user might be prompted to authenticate manually with their network credentials otherwise authentication fails the single sign-on (SSO) to the service providers.

Configuring Microsoft Internet Explorer

You can configure Internet Explorer to support user authentication when using a Kerberos Adapter instance.

About this task
To configure Internet Explorer, edit the settings in Internet Options.

Steps

1. Add the base URL to **Local intranet**.

   > ⓘ **Note**:
   >
   > Skip this step if the base URL *<pf-idp.domain.name>* is internal and not fully qualified. For example, if the base URL is `pingfederate`, you can skip this step. However, if *<pf-idp.domain.name>* is `www.example.com`, then you must add the base URL to the **Sites** list, as described in the following sub steps.

   a. Close all Internet Explorer tabs and windows.
   b. Open **Control Panel**# **Internet Options**.
   c. Click the **Security** tab.
   d. Select **Local intranet**, and then click **Sites**.
   e. Click **Advanced**.
   f. Enter the base URL. For example, `www.example.com`, and then click **Add**.
   g. Click **Close**. Click **OK** to return to the **Security** tab.

2. Verify **Automatic logon only in the Intranet zone** is selected.

   a. Under the **Security** tab, select **Local intranet** and then click **Custom level**.
   b. Verify **Automatic logon only in the Intranet zone** is selected in the **Settings** pane.
   c. Click **OK** to return to the **Security** tab.

3. Verify proxy settings.

> ⓘ **Note:**
>
> Skip the following sub steps if a proxy is not used.

  a. Click the **Connections** tab.
  b. Click **LAN settings**.
  c. Verify the **Use a proxy server for your LAN ...** check box is selected. Click **Advanced**.
  d. Enter the base URL in the **Exceptions** field. Click **OK**.
  e. Click **OK** to return to the **Connections** tab.
4. Verify **Enable Integrated Windows Authentication** is selected.
  a. Click the **Advanced** tab.
  b. Verify **Enable Integrated Windows Authentication** is selected in the **Settings** pane.
5. Click **OK** to close Internet Options.

Configuring Mozilla Firefox

You can configure Kerberos authentication using a Firefox browser.

Steps

1. Start Firefox.
2. Open a new tab, and then enter `about:config` in the address bar.
3. Double-click the `network.negotiate-auth.trusted-uris` preference name to modify its value to include the base URL of your PingFederate Bridge environment. For example, `www.example.com`.
4. Click **OK** and close the `about:config` tab.
5. Optional: Exit Firefox.

Index

A

attribute sources
    custom *129*

C

certificates
    revocation of *107*

D

datastore
    introduction *120*

P

pseudonyms
    unique values for *25*

U

unique values, for pseudonym creation *25*