



# **EBS Integration Kit Version 2.3.0.2**

---

Deployment Guide – Deploying the EIK in the PingFederate Server

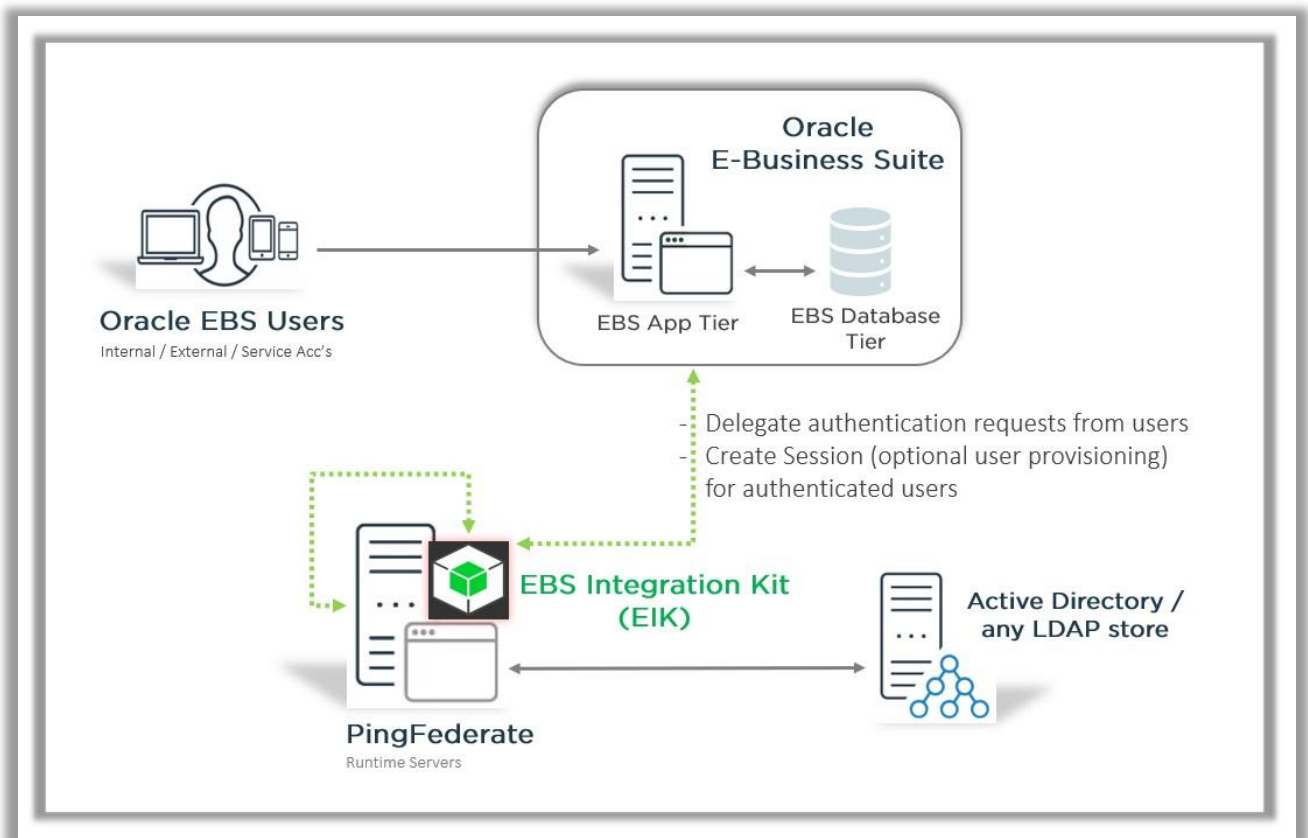
## Contents

1. EBS INTEGRATION KIT DESCRIPTION .....	3
2. FEATURES .....	4
3. BENEFITS .....	4
4. SYSTEM REQUIREMENTS .....	4
5. OVERVIEW OF THE FLOW .....	5
6. DEPLOYMENT AND CONFIGURATION OF PINGFEDERATE .....	5
6.1 Basic Configurations in Pingfederate for EIK .....	5
6.1.1 Configure HTML Form Adapter .....	5
6.1.2 OAuth Server side Configuration .....	7
6.2 Configure SSL Server certificate .....	18
7. EIK Deployment in PingFederate .....	18
7.1 EIKAuth Config File Generation .....	19
7.2 EBS DataSource (DBCX) File Creation .....	20
7.3 Deploying the log4j2.xml File .....	22
7.4 Deploying the cacerts File .....	22
7.5 Deploying the security.properties File .....	22
7.6 Deploying EBSAuth.war File .....	22
8. System Profile Parameter changes in EBS .....	23
8.1 System Profile Changes .....	23
9. SSO Testing for Oracle EBS .....	24

# 1. EBS INTEGRATION KIT DESCRIPTION

PingFederate as an Identity Federation product includes adapters and integration kits that simplify integrations with Identity Stores as well as Applications. The Oracle EBS solution uses these capabilities to enable Single Sign-On. The EBS INTEGRATION KIT (EIK) from LikeMinds Consulting provides a simple deployment path with PingFederate to enable single sign-on (SSO) to Oracle E-Business Suite.

## How does the EBS Integration Kit work?



The EBS Integration Kit is a light-weight java application which can be deployed in the PingFederate jetty-container which eliminates the need for dedicated server/hardware. The EIK is OpenID Connect Compliant as a resource server and can be configured with PingFederate to receive Authentication and Authorization tokens. The EBS Integration Kit communicates with the Oracle E-Business Suite Database for Session creation and optional user-provisioning if required.

## 2. FEATURES

- Works seamlessly with PingFederate
- Eliminated the need for additional Oracle components like Oracle Internet Directory or EBS AccessGate
- Provides flexibility of Deployment Methods: The EIK can be installed on the PingFederate server or any other J2EE container
- Supports Just-In-Time provisioning

## 3. BENEFITS

- Helps achieve painless integration of modern IAM into your existing architecture
- Enables you to modernize or migrate off legacy IAM without disrupting critical access to Oracle E-Business Suite
- Frees you from vendor lock-in with legacy IAM licensing—you will only require your EBS license
- Requires no provisioning of additional hardware components

## 4. SYSTEM REQUIREMENTS

- Java JDK 1.8 (Recommended Version)
- PingFederate 8 and above
- EBS IKT package with a valid license file.

EBS Integration Kit	Oracle E-Business Suite	Oracle Database	PingFederate	PingAccess (optional)	WebLogic	Apache Tomcat
EIK 2.3.0.2	Releases with update packs 12.0.x to 12.1.x & 12.2 to 12.2.10	From 11g up to 19c release	PingFederate versions 8.x to 11.0.2	PingAccess versions 4.x to 7.0.x	WebLogic 12.2.1.4	Apache Tomcat 9.0.65

## 5. OVERVIEW OF THE FLOW

1. When a user tries to login, his/her credentials are sent to Pingfederate.
2. Pingfederate validates the entered credentials against the configured Authentication Store (i.e., any LDAP directory).
3. The existence of the authenticated user will be checked against the EBS Database. If the user does exist in the EBS DB, an EBS session would be created, and the user would be allowed access to the requested page in EBS.
4. If the authenticated user does not exist in the EBS DB, the user would not be authorized to access the EBS Application.
5. If Just-In-Time Provisioning is enabled, authenticated users not present in the EBS DB would be provisioned by the EBS IKT, and EBS Sessions would then be created, thereby giving access to the EBS Application.

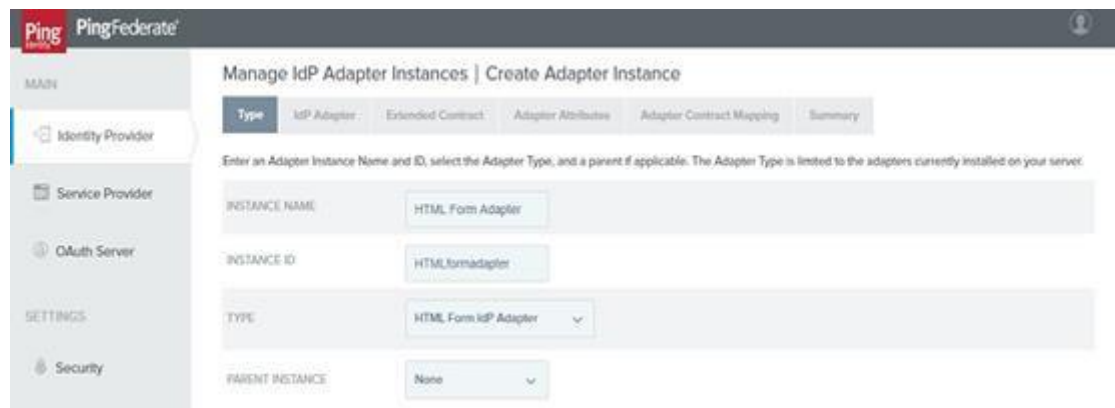
## 6. DEPLOYMENT AND CONFIGURATION OF PINGFEDERATE

PingFederate is a federation server that provides web single sign-on and API security for the resources in your Organization.

### 6.1 Basic Configurations in Pingfederate for EIK

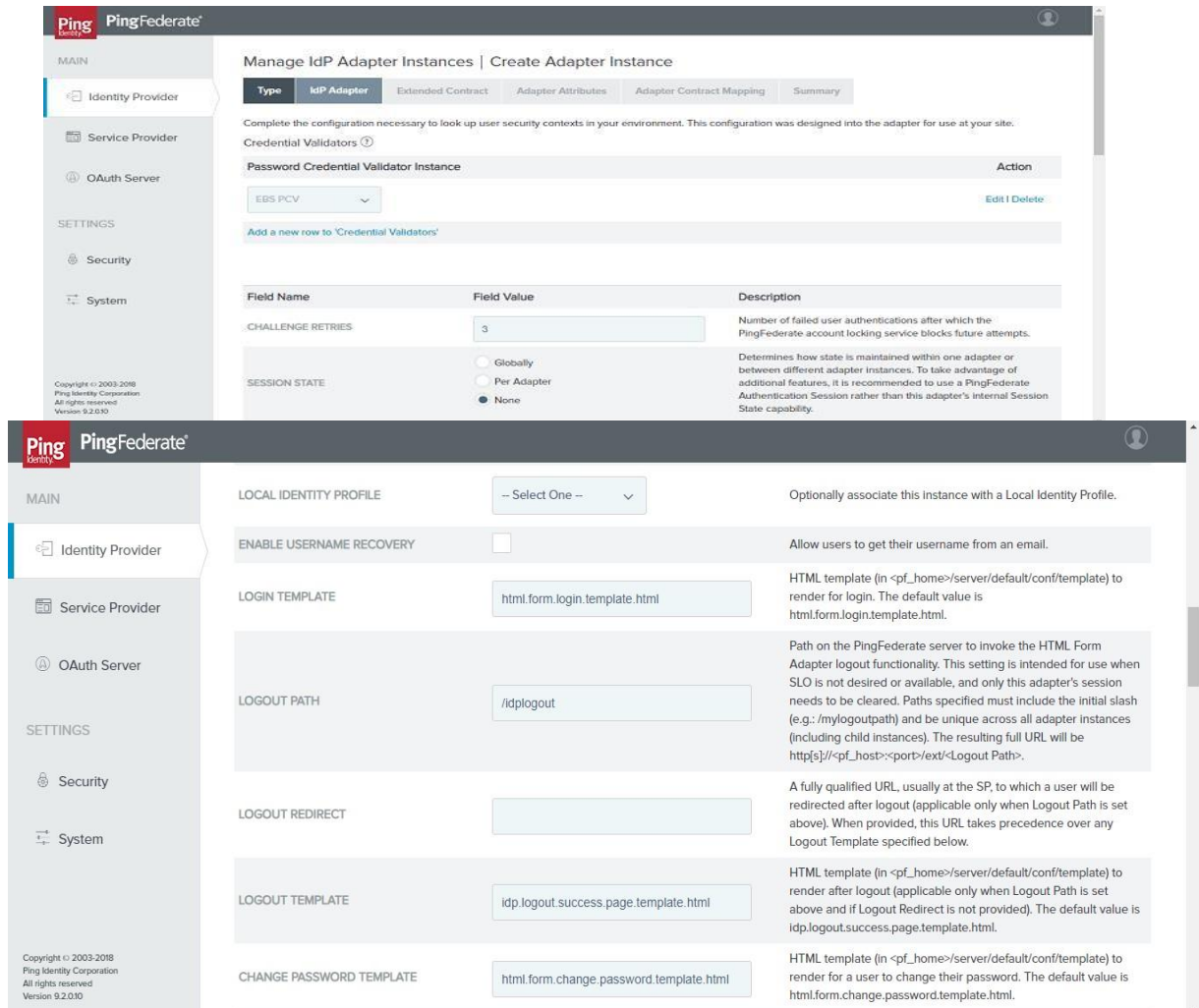
#### 6.1.1 Configure HTML Form Adapter

1. In PingFederate console go to Identity Provider >> Adapters>>Click New Adapter
2. Provide the Instance Name & ID then Select “Type” as “HTML Form IdP Adapter” and click Next.



The screenshot shows the PingFederate console interface. On the left is a navigation menu with options: MAIN, Identity Provider (selected), Service Provider, OAuth Server, SETTINGS, and Security. The main content area is titled 'Manage IdP Adapter Instances | Create Adapter Instance'. It features a tabbed interface with 'Type' selected. Below the tabs, there is a text prompt: 'Enter an Adapter Instance Name and ID, select the Adapter Type, and a parent if applicable. The Adapter Type is limited to the adapters currently installed on your server.' The form contains four fields: 'INSTANCE NAME' with the value 'HTML Form Adapter', 'INSTANCE ID' with the value 'HTMLformadapter', 'TYPE' with a dropdown menu showing 'HTML Form IdP Adapter', and 'PARENT INSTANCE' with a dropdown menu showing 'None'.

3. In the Next page, Add the “Password Credentials Validator” and click “Show advanced field” to add “/idplogout” in Logout Path Text Field. Then Click Next.



**PingFederate**

Manage IdP Adapter Instances | Create Adapter Instance

**Identity Provider**

Service Provider

OAuth Server

SETTINGS

Security

System

Copyright © 2003-2018 Ping Identity Corporation All rights reserved Version 9.2.0.10

Complete the configuration necessary to look up user security contexts in your environment. This configuration was designed into the adapter for use at your site.

**Credential Validators**

**Password Credential Validator Instance**

EBS PCV

Edit | Delete

Add a new row to 'Credential Validators'

Field Name	Field Value	Description
CHALLENGE RETRIES	3	Number of failed user authentications after which the PingFederate account locking service blocks future attempts.
SESSION STATE	<input type="radio"/> Globally <input type="radio"/> Per Adapter <input checked="" type="radio"/> None	Determines how state is maintained within one adapter or between different adapter instances. To take advantage of additional features, it is recommended to use a PingFederate Authentication Session rather than this adapter's internal Session State capability.

**LOCAL IDENTITY PROFILE** -- Select One --

Optionally associate this instance with a Local Identity Profile.

**ENABLE USERNAME RECOVERY** ☐ Allow users to get their username from an email.

**LOGIN TEMPLATE** html.form.login.template.html

HTML template (in <pf\_home>/server/default/conf/template) to render for login. The default value is html.form.login.template.html.

**LOGOUT PATH** /idplogout

Path on the PingFederate server to invoke the HTML Form Adapter logout functionality. This setting is intended for use when SLO is not desired or available, and only this adapter's session needs to be cleared. Paths specified must include the initial slash (e.g.: /mylogoutpath) and be unique across all adapter instances (including child instances). The resulting full URL will be http[s]://<pf\_host>:<port>/ext/<Logout Path>.

**LOGOUT REDIRECT**

A fully qualified URL, usually at the SP, to which a user will be redirected after logout (applicable only when Logout Path is set above). When provided, this URL takes precedence over any Logout Template specified below.

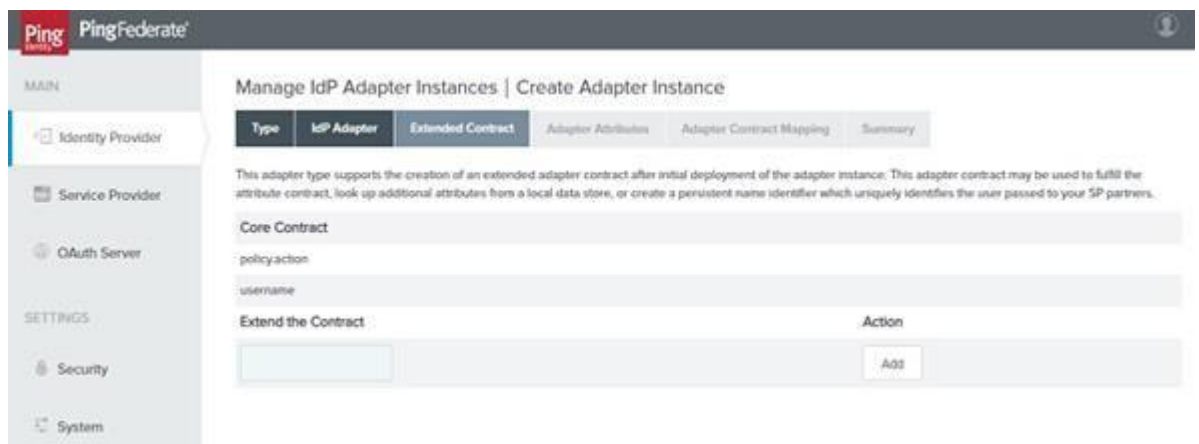
**LOGOUT TEMPLATE** idp.logout.success.page.template.html

HTML template (in <pf\_home>/server/default/conf/template) to render after logout (applicable only when Logout Path is set above and if Logout Redirect is not provided). The default value is idp.logout.success.page.template.html.

**CHANGE PASSWORD TEMPLATE** html.form.change.password.template.html

HTML template (in <pf\_home>/server/default/conf/template) to render for a user to change their password. The default value is html.form.change.password.template.html.

4. Add any additional attribute in Extended Contracts if required or leave the default values as such.



**PingFederate**

Manage IdP Adapter Instances | Create Adapter Instance

**Identity Provider**

Service Provider

OAuth Server

SETTINGS

Security

System

Copyright © 2003-2018 Ping Identity Corporation All rights reserved Version 9.2.0.10

This adapter type supports the creation of an extended adapter contract after initial deployment of the adapter instance. This adapter contract may be used to fulfill the attribute contract, look up additional attributes from a local data store, or create a persistent name identifier which uniquely identifies the user passed to your SP partners.

**Core Contract**

policy.action

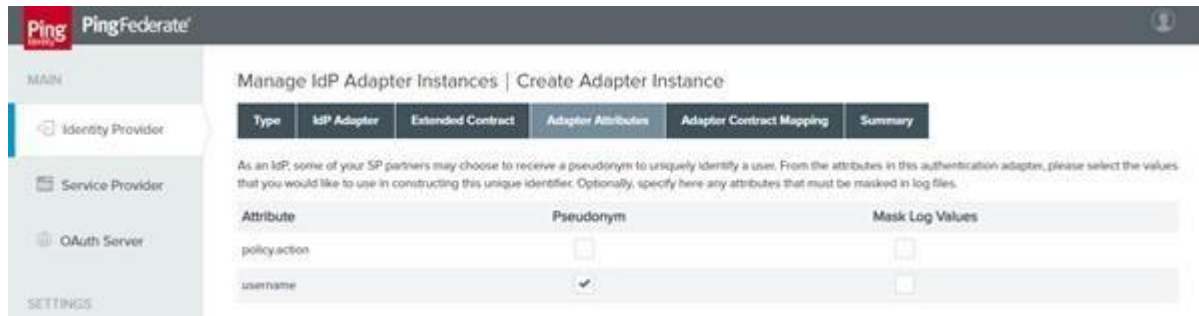
username

**Extend the Contract**

Action

Add

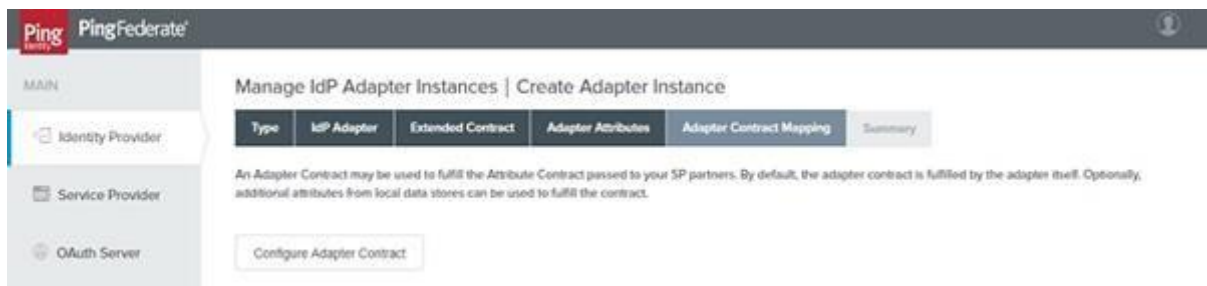
5. Enable “Pseudonym” for username (SP Partners may choose to receive so that they can uniquely identify a user) and click Next.



The screenshot shows the 'Manage IdP Adapter Instances' page in PingFederate, specifically the 'Adapter Attributes' tab. The left sidebar contains navigation links for MAIN, Identity Provider, Service Provider, OAuth Server, and SETTINGS. The main content area has a header 'Manage IdP Adapter Instances | Create Adapter Instance' and a sub-header with tabs: Type, IdP Adapter, Extended Contract, Adapter Attributes (selected), Adapter Contract Mapping, and Summary. Below the tabs, a text block explains that as an IdP, some SP partners may choose to receive a pseudonym to uniquely identify a user. A table follows with columns 'Attribute', 'Pseudonym', and 'Mask Log Values'. The 'username' attribute has the 'Pseudonym' checkbox checked.

Attribute	Pseudonym	Mask Log Values
policyaction	<input type="checkbox"/>	<input type="checkbox"/>
username	<input checked="" type="checkbox"/>	<input type="checkbox"/>

6. Configure the Adapter Contract Mapping or leave it as default and click Next.



The screenshot shows the 'Manage IdP Adapter Instances' page in PingFederate, specifically the 'Adapter Contract Mapping' tab. The left sidebar is the same as the previous screenshot. The main content area has the same header and sub-header tabs. Below the tabs, a text block explains that an Adapter Contract may be used to fulfill the Attribute Contract passed to your SP partners. A 'Configure Adapter Contract' button is visible at the bottom.

7. Review the Summary and Click Done and proceed to Save the adapter.

### 6.1.2 OAuth Server side Configuration

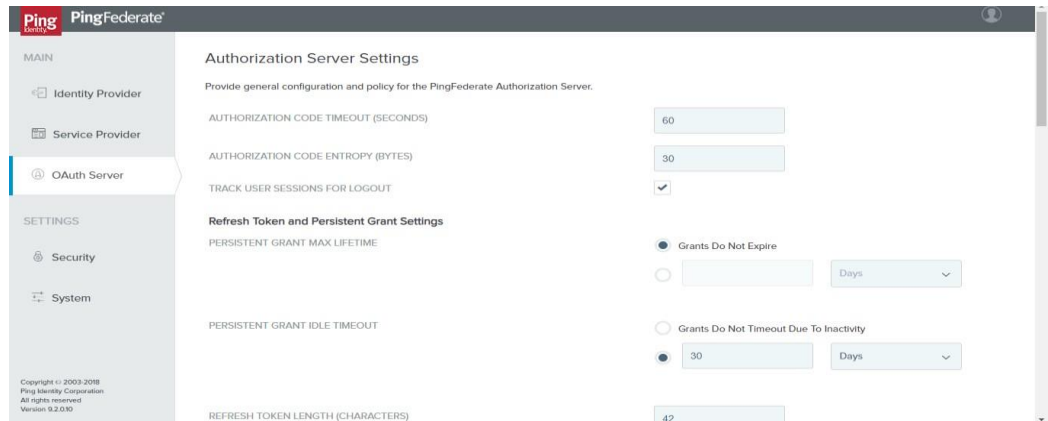
Configuring OAuth/OIDC flow PingFederate & EBS Integration Kit requires the following sections to be configured.

- Authorization server settings
- Scope Management
- IDP Adapter Mapping
- Access Token Management
- Access Token Mapping
- OpenID Connect Policy Management.
- Client Creation

### 6.1.2.1 Authorization Server Settings

Authorization Server Settings provide general configuration and policy for the PingFederate Authorization Server.

1. Go to OAuth Server >> Under Authorization Server >> Authorization Server Settings.
2. Enable the Track Users session for Log out checkbox and leave the remaining as default and click Save.

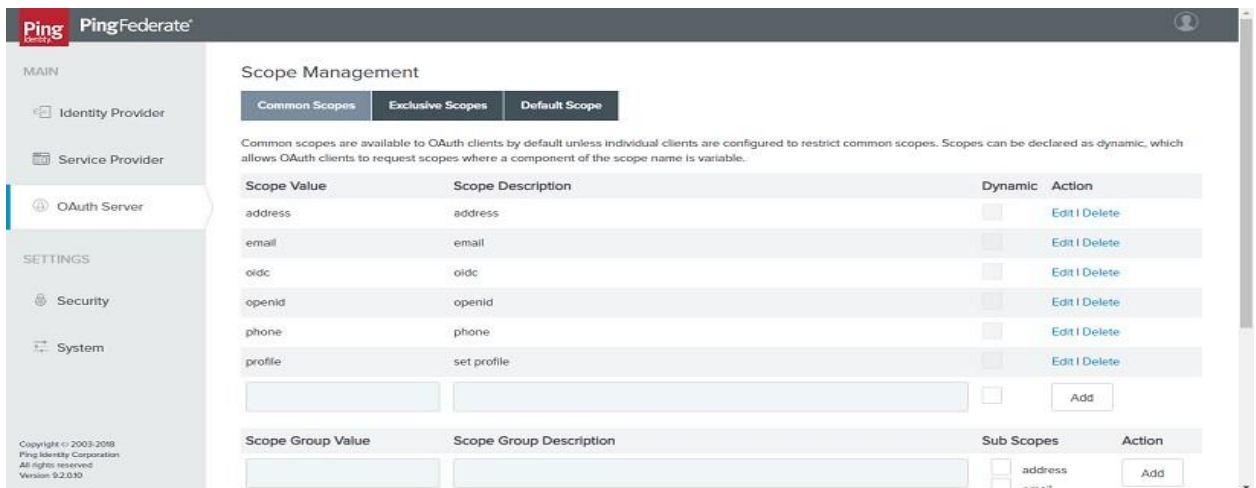


The screenshot shows the 'Authorization Server Settings' page in the PingFederate console. The left sidebar has a 'MAIN' section with 'Identity Provider', 'Service Provider', and 'OAuth Server' (selected), and a 'SETTINGS' section with 'Security' and 'System'. The main content area is titled 'Authorization Server Settings' and includes a description: 'Provide general configuration and policy for the PingFederate Authorization Server.' The settings include: 'AUTHORIZATION CODE TIMEOUT (SECONDS)' set to 60, 'AUTHORIZATION CODE ENTROPY (BYTES)' set to 30, 'TRACK USER SESSIONS FOR LOGOUT' checked, 'Refresh Token and Persistent Grant Settings' section with 'PERSISTENT GRANT MAX LIFETIME' set to 'Grants Do Not Expire' and 'PERSISTENT GRANT IDLE TIMEOUT' set to 'Grants Do Not Timeout Due To Inactivity' with a value of 30 days, and 'REFRESH TOKEN LENGTH (CHARACTERS)' set to 42.

### 6.1.2.2 Scope Management

Scopes can be declared as dynamic, which allows OAuth clients to request scopes where a component of the scope name is variable. Common scopes are available to OAuth clients by default unless individual clients are configured to restrict common scopes.

1. Go to OAuth Server >> Under Authorization Server >> Scope Management
2. These are the Default scopes for PingFederate
  - a) Email
  - b) Profile
  - c) Openid
  - d) Address
  - e) Phone



The screenshot shows the 'Scope Management' page in the PingFederate console. The left sidebar is the same as the previous screenshot. The main content area is titled 'Scope Management' and has three tabs: 'Common Scopes', 'Exclusive Scopes', and 'Default Scope' (selected). Below the tabs is a description: 'Common scopes are available to OAuth clients by default unless individual clients are configured to restrict common scopes. Scopes can be declared as dynamic, which allows OAuth clients to request scopes where a component of the scope name is variable.' Below this is a table of common scopes:

Scope Value	Scope Description	Dynamic	Action
address	address	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
email	email	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
oidc	oidc	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
openid	openid	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
phone	phone	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
profile	set profile	<input type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>

Below the table are input fields for 'Scope Value', 'Scope Description', and 'Dynamic' (checkbox), followed by an 'Add' button. Below this is a section for 'Scope Group Value' and 'Scope Group Description', followed by a 'Sub Scopes' section with a table:

Scope Group Value	Scope Group Description	Sub Scopes	Action
		address email	<a href="#">Add</a>



### 6.1.2.3 IdP Adapter Mapping

This configuration allows you to map attributes based on an IdP adapter configuration into the USER\_KEY and USER\_NAME attributes (presented to the user for authorization permission) for a persistent grant, as well as the extended attributes.

1. Go to OAuth Server >> Authorization Server >> IdP Adapter Mapping and select HTML Form Adapter and click Add Mapping.
2. Add any Data Store details if values are pulled from a Data Store. If not, click Next.
3. In Contract Fulfillment, Map the USER\_NAME & USER\_KEY value from either the Adapter or Data Store.



**PingFederate**

IdP Adapter Mappings | IdP Adapter Mapping

Attribute Sources & User Lookup | **Contract Fulfillment** | Issuance Criteria | Summary

Select a Source and Value to map into each item in the Contract list.

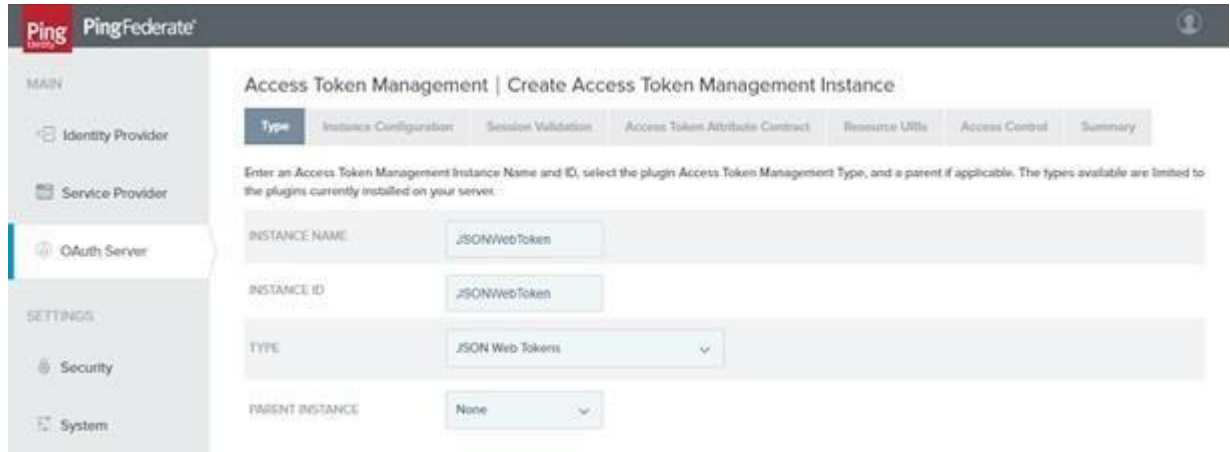
Contract	Source	Value	Actions
USER_KEY	Adapter	username	None available
USER_NAME	Adapter	username	None available

4. In the next page, provide any access restriction criteria details if the restriction of user access is required or click Next.
5. Review the Summary and then click Done and Save.

### 6.1.2.4 Access Token Management

PingFederate uses Access Token Management plugins to issue and validate OAuth access tokens. Each plugin instance can have its own token type, configuration settings, and attribute contract.

1. Go to OAuth Server >> Under Token Mapping >> Access Token Management >> Create New Instance by providing the Instance Name & ID, Type as “JSON Web Token” Click Next.



**Ping Federate** Access Token Management | Create Access Token Management Instance

**Type** Instance Configuration Session Validation Access Token Attribute Contract Resource URIs Access Control Summary

Enter an Access Token Management Instance Name and ID, select the plugin Access Token Management Type, and a parent if applicable. The types available are limited to the plugins currently installed on your server.

INSTANCE NAME: JSONWebToken

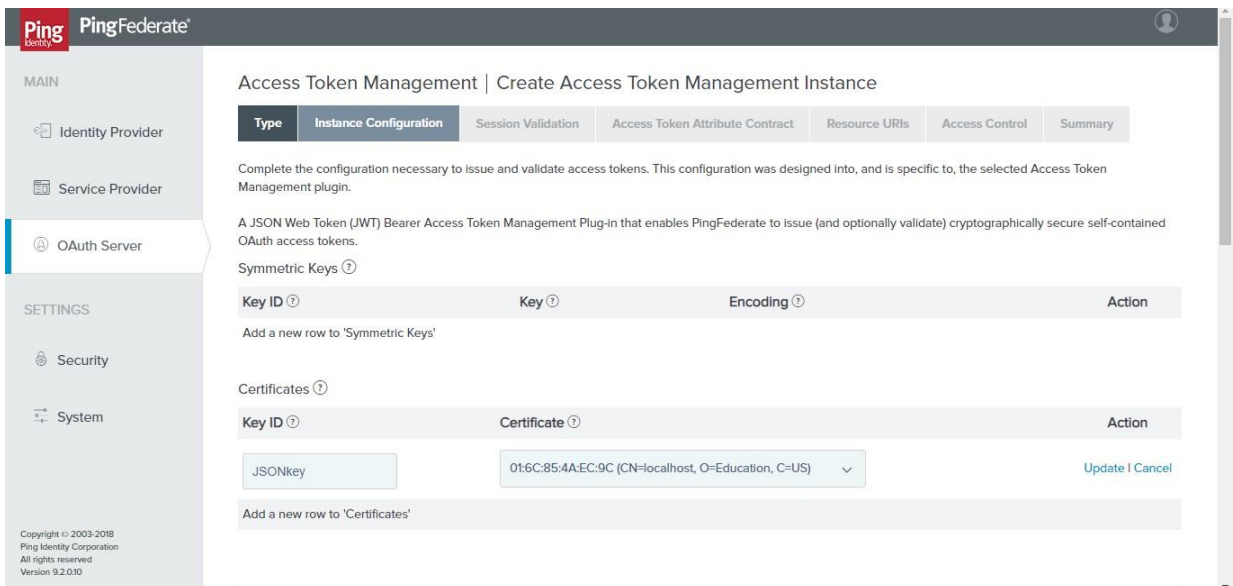
INSTANCE ID: JSONWebToken

TYPE: JSON Web Tokens

PARENT INSTANCE: None

2. In Instance Configuration Section, under Certificates, Add the Signing & Decryption certificate.

Note: When defining an access token management instance, you can customize various settings, including the token format, lifetime, and attribute contract for this instance.



**Ping Federate** Access Token Management | Create Access Token Management Instance

**Type** Instance Configuration Session Validation Access Token Attribute Contract Resource URIs Access Control Summary

Complete the configuration necessary to issue and validate access tokens. This configuration was designed into, and is specific to, the selected Access Token Management plugin.

A JSON Web Token (JWT) Bearer Access Token Management Plug-in that enables PingFederate to issue (and optionally validate) cryptographically secure self-contained OAuth access tokens.

Symmetric Keys

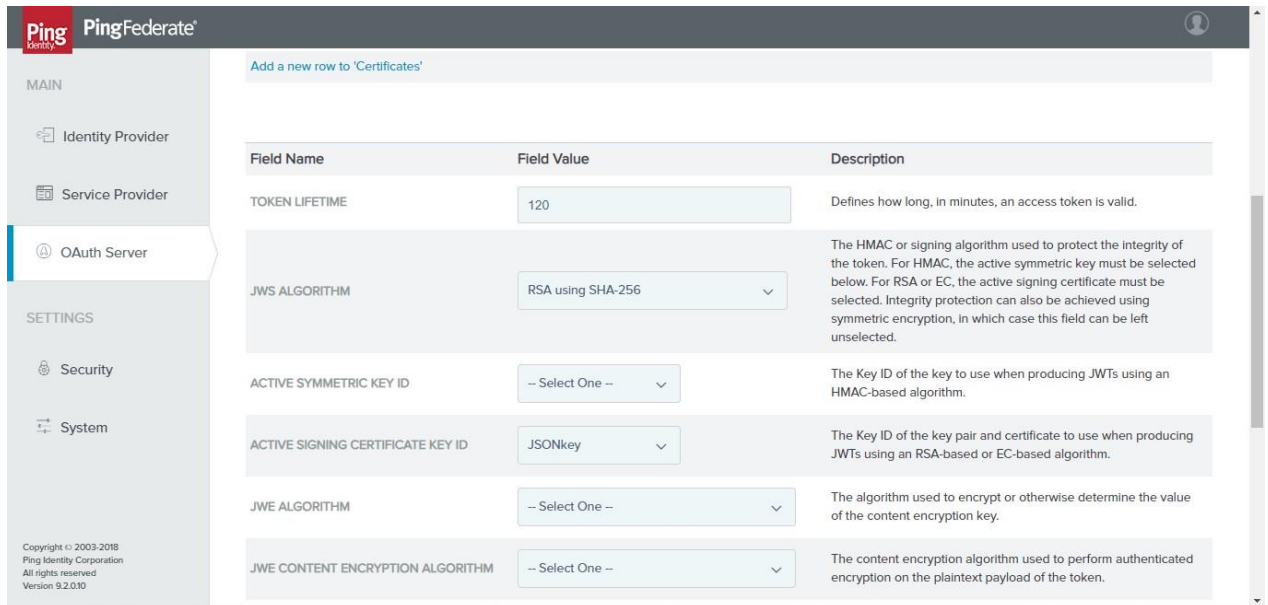
Key ID	Key	Encoding	Action
Add a new row to 'Symmetric Keys'			

Certificates

Key ID	Certificate	Action
JSONkey	016C:85:4A:EC:9C (CN=localhost, O=Education, C=US)	Update   Cancel
Add a new row to 'Certificates'		

Copyright © 2003-2018  
Ping Identity Corporation  
All rights reserved  
Version 9.2.0.10

- Under JWS Algorithm Dropdown, choose RSA using SHA-256.
- Under Active Signing Certificate Key ID Dropdown, select the key ID which is added above.



**PingFederate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

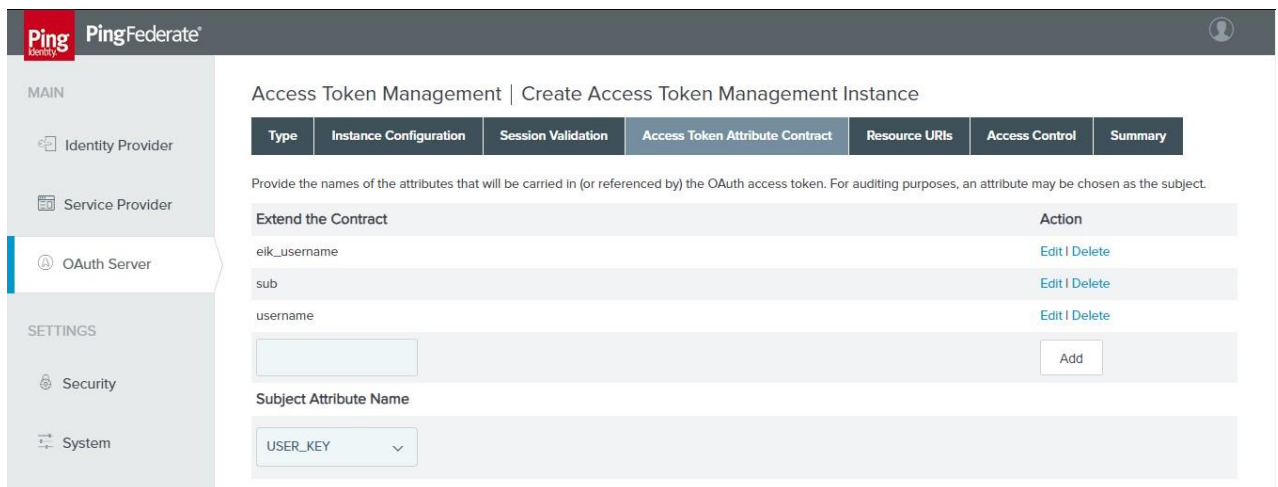
- Security**
- System

Copyright © 2003-2018  
 Ping Identity Corporation  
 All rights reserved  
 Version 9.2.0.10

Add a new row to 'Certificates'

Field Name	Field Value	Description
TOKEN LIFETIME	120	Defines how long, in minutes, an access token is valid.
JWS ALGORITHM	RSA using SHA-256	The HMAC or signing algorithm used to protect the integrity of the token. For HMAC, the active symmetric key must be selected below. For RSA or EC, the active signing certificate must be selected. Integrity protection can also be achieved using symmetric encryption, in which case this field can be left unselected.
ACTIVE SYMMETRIC KEY ID	-- Select One --	The Key ID of the key to use when producing JWTs using an HMAC-based algorithm.
ACTIVE SIGNING CERTIFICATE KEY ID	JSONkey	The Key ID of the key pair and certificate to use when producing JWTs using an RSA-based or EC-based algorithm.
JWE ALGORITHM	-- Select One --	The algorithm used to encrypt or otherwise determine the value of the content encryption key.
JWE CONTENT ENCRYPTION ALGORITHM	-- Select One --	The content encryption algorithm used to perform authenticated encryption on the plaintext payload of the token.

- In the next page, there will be no changes to the Session Validation Settings here. So, leave it defaults.
- In the below screenshot, In Access Token Attribute Contract, we define the attribute contract for the access tokens issued by this access token management instance. Extend the contract with the 'eik\_username' & 'username' attributes.



**PingFederate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

- Security**
- System

Access Token Management | Create Access Token Management Instance

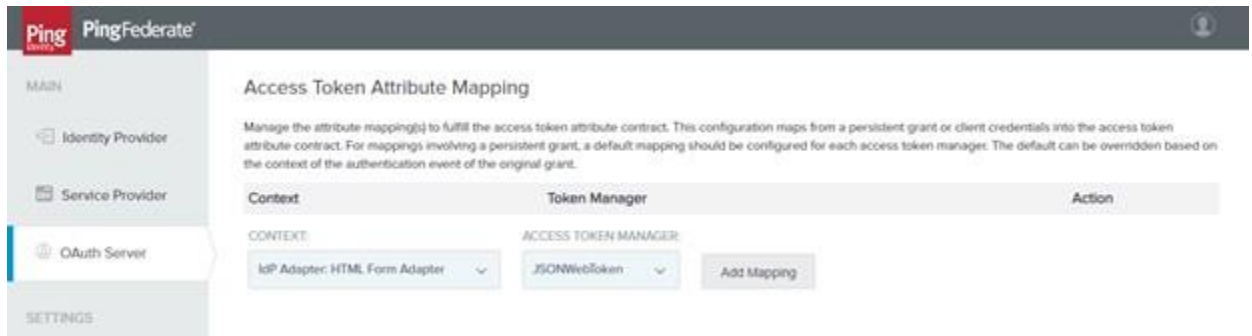
Type	Instance Configuration	Session Validation	Access Token Attribute Contract	Resource URIs	Access Control	Summary
Provide the names of the attributes that will be carried in (or referenced by) the OAuth access token. For auditing purposes, an attribute may be chosen as the subject.						
<b>Extend the Contract</b>						<b>Action</b>
eik_username						Edit   Delete
sub						Edit   Delete
username						Edit   Delete
<input type="text"/>						Add
<b>Subject Attribute Name</b>						
USER_KEY						

- Click Next.
- Under Access Control, we can restrict the clients who can use this Access Token Management Instance, which we do not use here. So, click Next.
- Review the Summary and Click Save.

### 6.1.2.5 Access Token Mapping

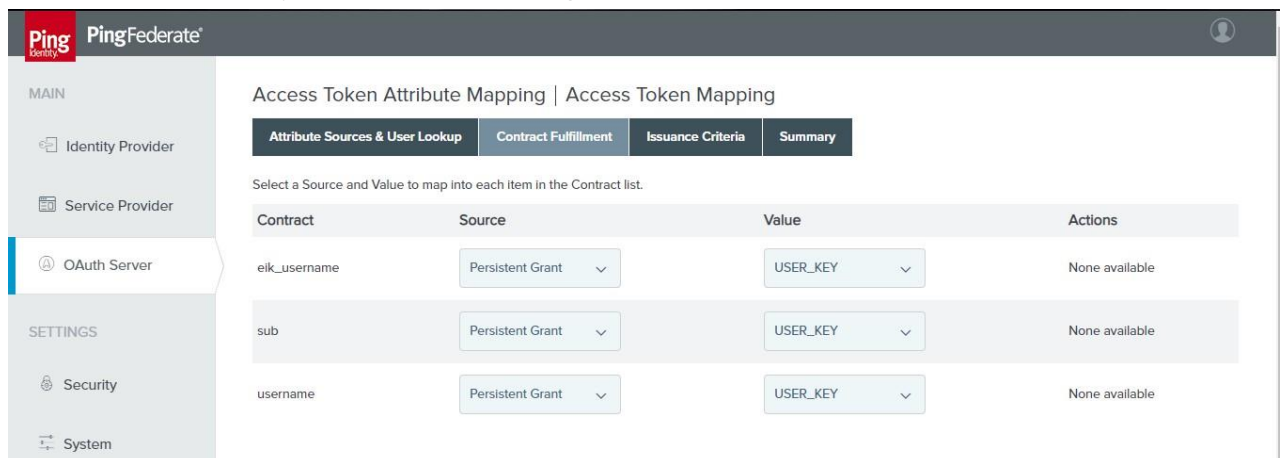
Access Token Mapping is used to map the attributes to fulfill the access token attribute contract. This configuration maps from a persistent grant into the access token attribute contract and there should be a default mapping configured for each access token manager.

1. Go to OAuth Server >> Token Mapping >> Access Token Mapping.
2. Map the “Context” (IDP Adapter: HTMLFormAdapter) to “Access Token Manager” (JSONWebToken) and then click “Add Mapping”.



3. In “Attribute Source Lookup” Tab (optional), we can add any Attribute Source to fetch the attribute values from any of your desired datastore and on completion click Next.
4. In “Contract Fulfillment” tab, map the Source as Persistent Grant and its value as “USER\_KEY” for all contracts as shown in the below screenshot.

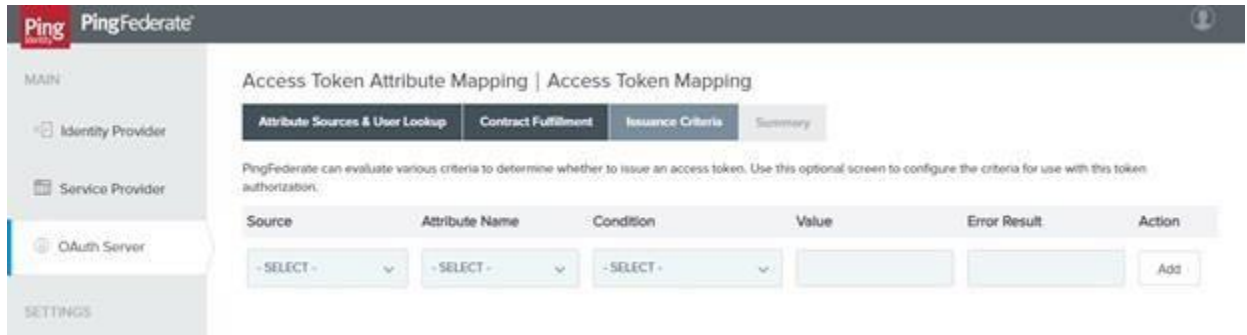
Note: Usually in the Contract Fulfillment screen, we map values into the token attribute contract. These are the attributes that will be included or referenced in the access token. When we select Persistent Grant, the associated Value list is populated by the USER\_KEY and extended attributes from the persistent access-token grant.



Contract	Source	Value	Actions
elik_username	Persistent Grant	USER_KEY	None available
sub	Persistent Grant	USER_KEY	None available
username	Persistent Grant	USER_KEY	None available

5. In the next page, provide any restriction rules to limit the unauthorized access or leave defaults & click Next.

Note: In this screen, we define the criteria that must be satisfied so that PingFederate processes the request further. In essence, this *token authorization* feature provides the capability to conditionally approve or reject requests based on individual attributes.



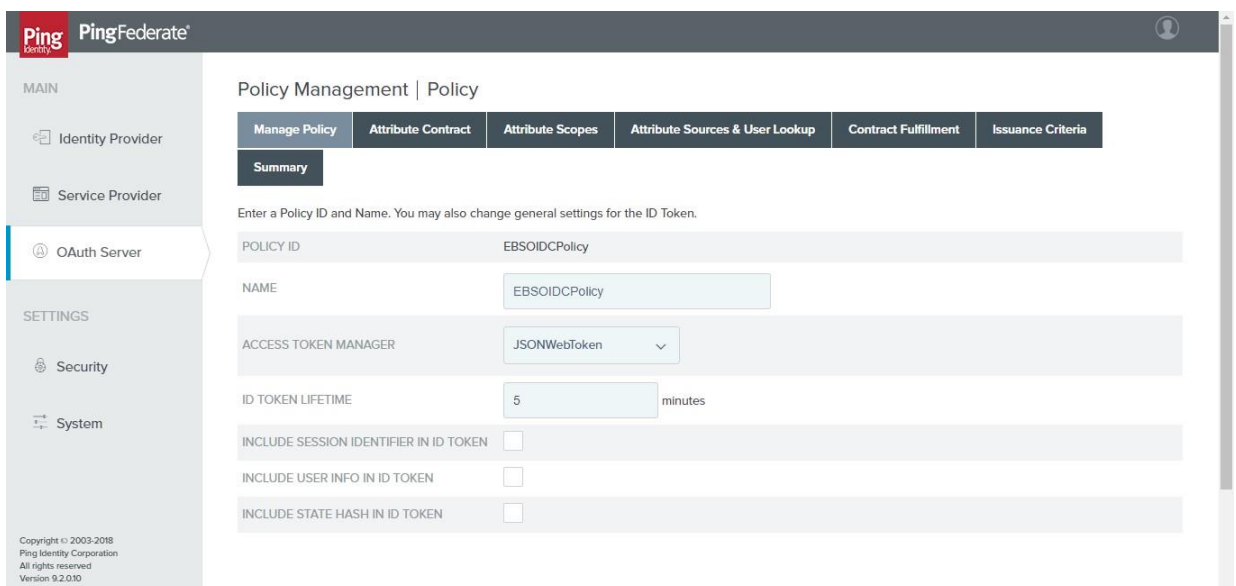
The screenshot shows the 'Access Token Attribute Mapping' screen in PingFederate. The left sidebar has 'OAuth Server' selected under the 'MAIN' section. The main content area has tabs for 'Attribute Sources & User Lookup', 'Contract Fulfillment', 'Issuance Criteria', and 'Summary'. The 'Issuance Criteria' tab is active. Below the tabs, there is a table with columns: Source, Attribute Name, Condition, Value, Error Result, and Action. Each of the first four columns has a dropdown menu with '- SELECT -' as the selected option. There is an 'Add' button at the end of the table.

6. Review the Summary and tap Save.

### 6.1.2.6 OpenID Connect Policy Management

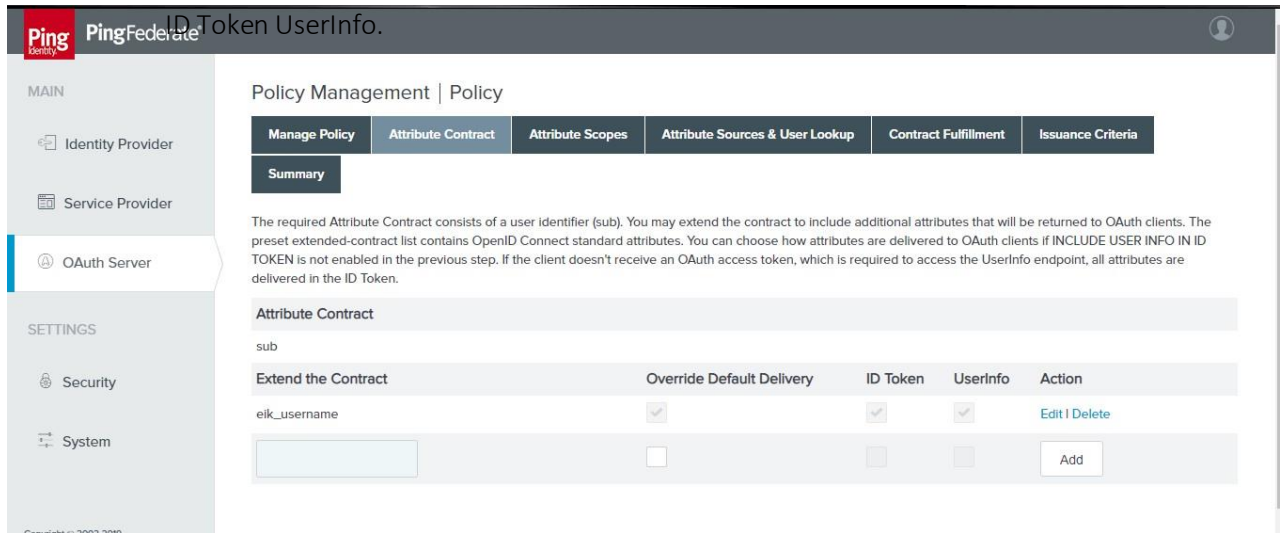
Here we define OpenID Connect policies and manage them for obtaining user attributes which act as the claims to be sent in an ID Token and, also in response to requests received at the PingFederate User Info endpoint. Policies that are defined here can be mapped to specific OAuth clients.

1. Go to OAuth Server >> Token Mapping >> OpenID Connect Policy Management >> Add Policy.
2. Provide a Policy Name & Select the Access Token Manager and leave the remaining as default. Click Next.



The screenshot shows the 'Policy Management' screen in PingFederate. The left sidebar has 'OAuth Server' selected under the 'MAIN' section. The main content area has tabs for 'Manage Policy', 'Attribute Contract', 'Attribute Scopes', 'Attribute Sources & User Lookup', 'Contract Fulfillment', and 'Issuance Criteria'. The 'Manage Policy' tab is active. Below the tabs, there is a 'Summary' section. It contains a form with the following fields: 'POLICY ID' (EBSOIDCPolicy), 'NAME' (EBSOIDCPolicy), 'ACCESS TOKEN MANAGER' (JSONWebToken), 'ID TOKEN LIFETIME' (5 minutes), 'INCLUDE SESSION IDENTIFIER IN ID TOKEN' (checkbox), 'INCLUDE USER INFO IN ID TOKEN' (checkbox), and 'INCLUDE STATE HASH IN ID TOKEN' (checkbox). At the bottom left, there is a copyright notice: 'Copyright © 2003-2018 Ping Identity Corporation. All rights reserved. Version 9.2.0.10'.

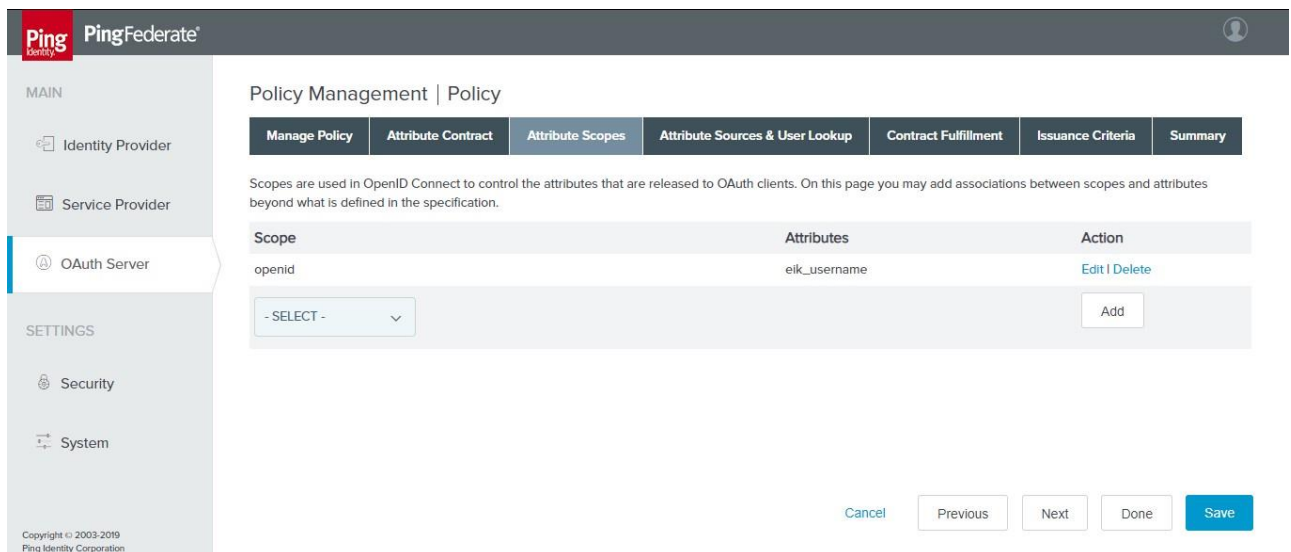
- Under Attribute Contract, Extend the Contract by manually adding 'eik\_username' and delete the other attributes present. Enable checkboxes for 'Override Default Delivery, ID Token UserInfo.



The screenshot shows the PingFederate web interface. The left sidebar has a 'MAIN' section with 'Identity Provider', 'Service Provider', and 'OAuth Server' (selected), and a 'SETTINGS' section with 'Security' and 'System'. The main content area is titled 'Policy Management | Policy' and has tabs for 'Manage Policy', 'Attribute Contract' (selected), 'Attribute Scopes', 'Attribute Sources & User Lookup', 'Contract Fulfillment', and 'Issuance Criteria'. Below the tabs is a 'Summary' section with a text block explaining the required Attribute Contract. Below that is an 'Attribute Contract' section with a table:

Extend the Contract	Override Default Delivery	ID Token	UserInfo	Action
eik_username	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#">Edit</a>   <a href="#">Delete</a>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Add"/>

- Click Next.
- Under Attribute Scopes Section, select openid from the Scope dropdown and in Attributes tick the 'eik\_username' checkbox.

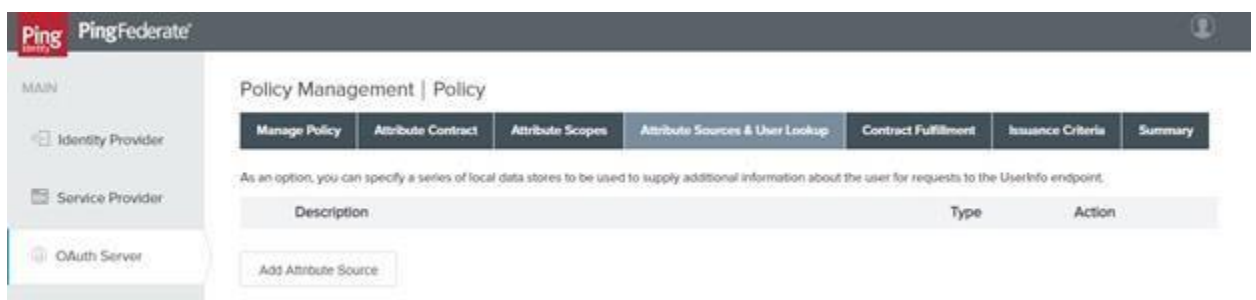


The screenshot shows the PingFederate web interface. The left sidebar is the same as the previous screenshot. The main content area is titled 'Policy Management | Policy' and has tabs for 'Manage Policy', 'Attribute Contract', 'Attribute Scopes' (selected), 'Attribute Sources & User Lookup', 'Contract Fulfillment', 'Issuance Criteria', and 'Summary'. Below the tabs is a text block explaining that scopes are used in OpenID Connect to control attributes. Below that is a table:

Scope	Attributes	Action
openid	eik_username	<a href="#">Edit</a>   <a href="#">Delete</a>
- SELECT -		<input type="button" value="Add"/>

At the bottom right, there are buttons: 'Cancel', 'Previous', 'Next', 'Done', and 'Save'.

- Attributes can be fetched from any of the datastore at the runtime or leave it as to use the values presented in the Token.

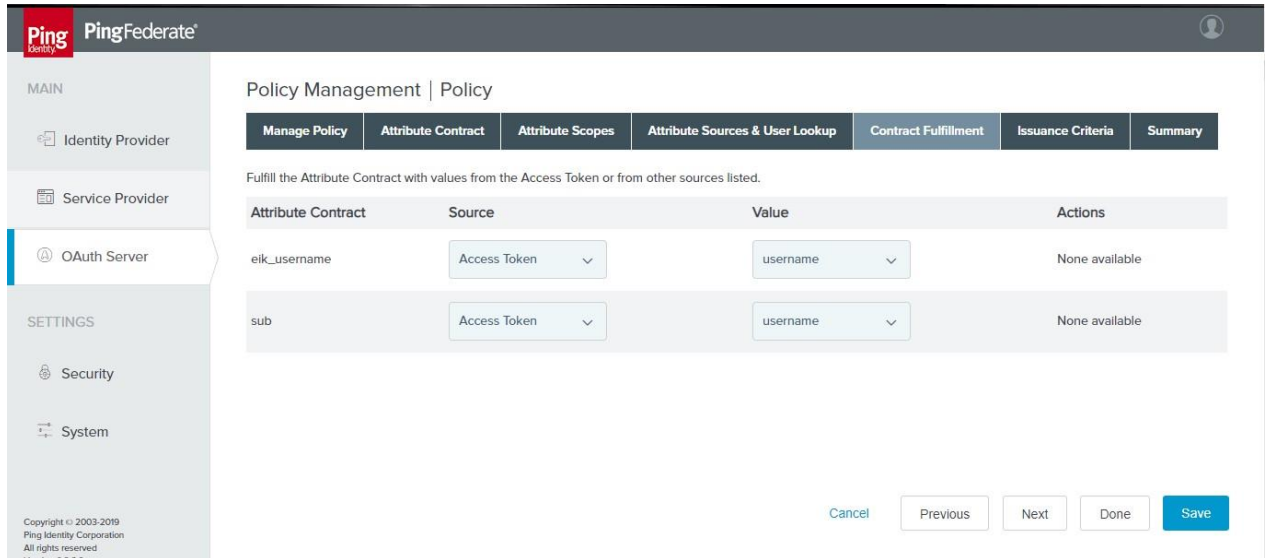


The screenshot shows the PingFederate web interface. The left sidebar is the same as the previous screenshots. The main content area is titled 'Policy Management | Policy' and has tabs for 'Manage Policy', 'Attribute Contract', 'Attribute Scopes', 'Attribute Sources & User Lookup' (selected), 'Contract Fulfillment', 'Issuance Criteria', and 'Summary'. Below the tabs is a text block explaining that as an option, you can specify a series of local data stores. Below that is a table:

Description	Type	Action
<input type="button" value="Add Attribute Source"/>		

- Under Contract Fulfillment Section, Map the Sub & eik\_username attribute from the source - Access Token and value – username.

Note: In the Contract Fulfillment screen, we map attributes from the access token or other sources to fulfill the attribute contract. Here, the value is provided from the access token.

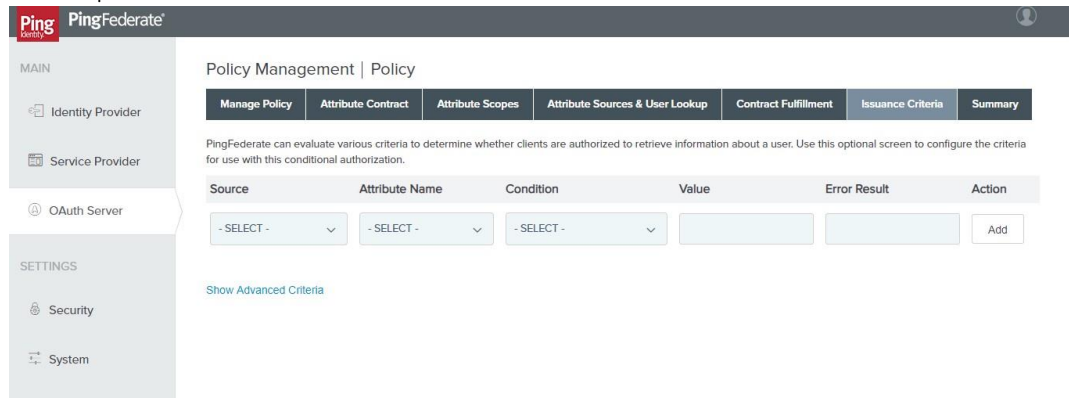


The screenshot shows the PingFederate Policy Management interface. The left sidebar has a 'MAIN' section with 'Identity Provider', 'Service Provider', and 'OAuth Server' (selected), and a 'SETTINGS' section with 'Security' and 'System'. The main content area is titled 'Policy Management | Policy' and has tabs for 'Manage Policy', 'Attribute Contract', 'Attribute Scopes', 'Attribute Sources & User Lookup', 'Contract Fulfillment' (active), 'Issuance Criteria', and 'Summary'. Below the tabs, a message states: 'Fulfill the Attribute Contract with values from the Access Token or from other sources listed.' A table follows with columns: 'Attribute Contract', 'Source', 'Value', and 'Actions'. It contains two rows: one for 'eik\_username' and one for 'sub', both with 'Access Token' as the source and 'username' as the value. The 'Actions' column for both rows says 'None available'. At the bottom right are buttons: 'Cancel', 'Previous', 'Next', 'Done', and 'Save'.

Attribute Contract	Source	Value	Actions
eik_username	Access Token	username	None available
sub	Access Token	username	None available

- In the next page, provide any restriction rules in case of restricting any unauthorized access.

Note: In this screen, we define the criteria that must be satisfied so that PingFederate processes the request further. In essence, this *token authorization* feature provides the capability to conditionally approve or reject requests based on individual attributes. We can define multiple criteria, so that in this case, *ALL* the criteria must be satisfied in order for PingFederate to move a request to the next phase.



The screenshot shows the PingFederate Policy Management interface, specifically the 'Issuance Criteria' tab. The left sidebar is the same as the previous screenshot. The main content area is titled 'Policy Management | Policy' and has tabs for 'Manage Policy', 'Attribute Contract', 'Attribute Scopes', 'Attribute Sources & User Lookup', 'Contract Fulfillment', 'Issuance Criteria' (active), and 'Summary'. Below the tabs, a message states: 'PingFederate can evaluate various criteria to determine whether clients are authorized to retrieve information about a user. Use this optional screen to configure the criteria for use with this conditional authorization.' A table follows with columns: 'Source', 'Attribute Name', 'Condition', 'Value', 'Error Result', and 'Action'. The 'Source', 'Attribute Name', and 'Condition' columns each have a '- SELECT -' dropdown menu. The 'Value' column has an empty text input field. The 'Error Result' column has an empty text input field. The 'Action' column has an 'Add' button. Below the table is a link that says 'Show Advanced Criteria'.

Source	Attribute Name	Condition	Value	Error Result	Action
- SELECT -	- SELECT -	- SELECT -			Add

- Review Summary and Click Save.

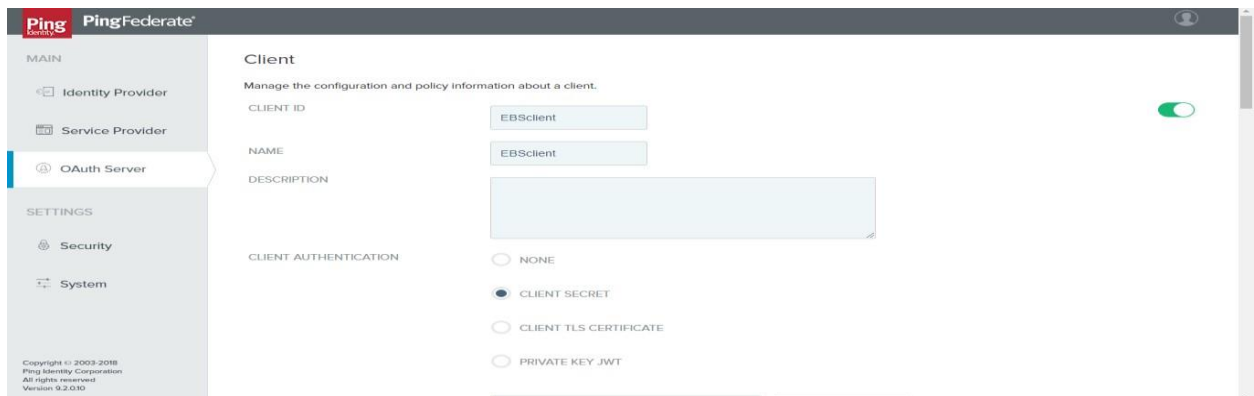


### 6.1.2.7 Client Configuration for OAuth Server

Here we can manage the configurations and policy information related to a client.

1. In the PingFederate Admin Console, go to OAuth Server >> Clients >> Create New
2. Create a new client with the following details
  - Client Name : "<Specify any Client Name for EIK>"
  - Client ID : "<Specify any Client ID for EIK>"
  - Client Secret : "<(Create secret using Generate Secret option)>"
3. Type the Redirection URL in the format:
 

https://<pf-hostname>:<pf-runtime-port>/<context>/handler
4. Tick the checkbox for Bypass Authorization Approval.
5. For Allowed Grant Types, select Authorization Code and Refresh Token.
6. For Default Access Token Manager choose JSONWebToken.
7. Under OpenID Connect section, choose the Policy you configured just before.
8. Leave the remaining as default and click Save.



**PingFederate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

- Security
- System

Copyright © 2003-2018 Ping Identity Corporation. All rights reserved. Version 9.2.0.30

#### Client

Manage the configuration and policy information about a client.

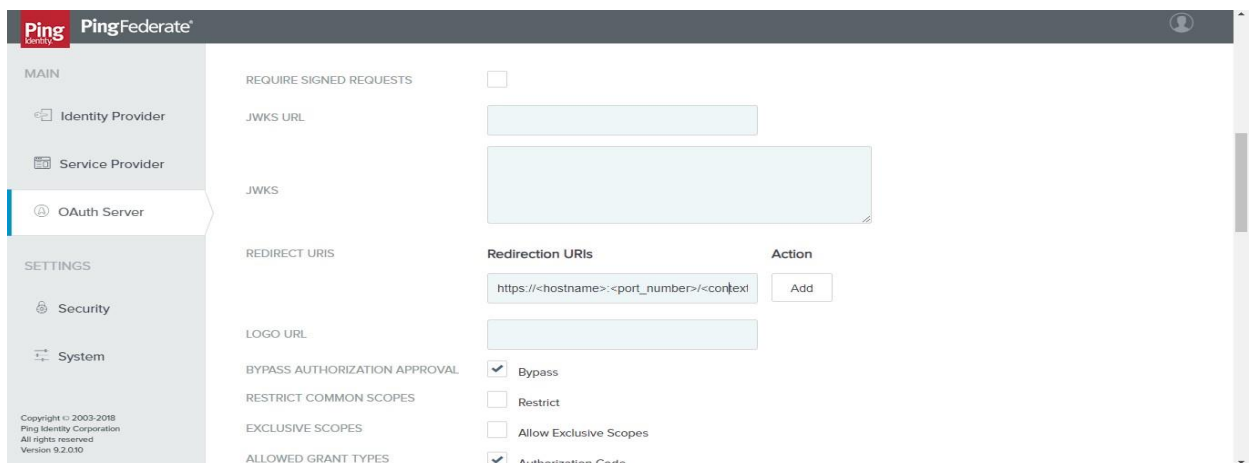
CLIENT ID: EBSClient

NAME: EBSClient

DESCRIPTION:

CLIENT AUTHENTICATION:

- ☐ NONE
- ☒ CLIENT SECRET
- ☐ CLIENT TLS CERTIFICATE
- ☐ PRIVATE KEY JWT



**PingFederate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

- Security
- System

Copyright © 2003-2018 Ping Identity Corporation. All rights reserved. Version 9.2.0.30

REQUIRE SIGNED REQUESTS: ☐

JWKS URL:

JWKS:

REDIRECT URIS:

Redirection URIs	Action
https://<hostname>:<port_number>/<context>/	Add

LOGO URL:

BYPASS AUTHORIZATION APPROVAL: ☒ Bypass

RESTRICT COMMON SCOPES: ☐ Restrict

EXCLUSIVE SCOPES: ☐ Allow Exclusive Scopes

ALLOWED GRANT TYPES: ☒ Authorization Code



**Ping Federate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

- Security
- System

Copyright © 2003-2009  
Ping Identity Corporation  
All rights reserved  
Version 9.2.0.0

BYPASS AUTHORIZATION APPROVAL ☒ Bypass

RESTRICT COMMON SCOPES ☐ Restrict

EXCLUSIVE SCOPES ☐ Allow Exclusive Scopes

ALLOWED GRANT TYPES

- ☒ Authorization Code
- ☐ Implicit
- ☐ Refresh Token
- ☐ Client Credentials
- ☐ Device Flow
- ☐ Resource Owner Password Credentials
- ☐ Extension Grants
- ☐ Access Token Validation (Client is a Resource Server)

RESTRICT RESPONSE TYPES ☐ Restrict

DEFAULT ACCESS TOKEN MANAGER **JSONWebToken**

VALIDATE AGAINST ALL ELIGIBLE ACCESS TOKEN MANAGERS ☐

PERSISTENT GRANTS MAX LIFETIME ☒ Use Global Setting

Activate Windows  
Go to PC settings to activate Windows.

**Ping Federate**

MAIN

- Identity Provider
- Service Provider
- OAuth Server**

SETTINGS

- Security
- System

Copyright © 2003-2018  
Ping Identity Corporation  
All rights reserved  
Version 9.2.0.10

PERSISTENT GRANTS IDLE TIMEOUT ☒ Use Global Setting

☐ Grants Do Not Timeout Due To Inactivity

☐  Days

REFRESH TOKEN ROLLING POLICY ☒ Use Global Setting ☐ Don't Roll ☐ Roll

OPENID CONNECT

ID Token Signing Algorithm **Default**

ID Token Key Management Encryption Algorithm **No Encryption**

Policy **EBSOIDCPOLICY**

☐ Grant Access to Session Revocation API

## 6.2 Configure SSL Server certificate

SSL certificates are created as they need to be presented for getting access to the PingFederate administrative console and for all the incoming HTTPS connections at the runtime engine nodes.

If you already own an SSL certificate for pingfederate we can skip to the Key Generation part, else we need to proceed with the following steps to configure a self-signed certificate:

1. In PingFederate Click Security >> Security and Certificate Key Management >> SSL Server Certificate >> Create New
2. Enter the details and create a certificate.
3. Export this certificate now with the option of exporting “Certificate Only” and not with the private key.

Note: As part of the Key Generation process, add the exported certificate to Trusted CAs in PingFederate Admin console.

## 7. EIK Deployment in PingFederate

The following steps should be followed for the deployment of EBS Integration Kit in PingFederate server. As a pre-requisite, the EIKAuth.config configuration file & EIK.dbcx EBS DataSource (DBCX) file should be generated before deploying the EIK war package.

Pre-Requirement: An environment variable ‘EIK\_HOME’ should be set in the PingFederate server before deploying the EBS Integration Kit.

1. Navigate to <pf\_install>/pingfederate directory
2. Create a new directory named “EBSAuth”. After creation, the directory structure will look like the following:  
    <pf\_install>/pingfederate/EBSAuth
3. Now, Place the EBS IKT license file “ebsauth.lic” in <pf\_install>/pingfederate /EBSAuth directory.
4. Now navigate to user’s base path using cd ~ and set the environmental variable for EIK\_HOME in .bash profile : vi .bash\_profile
5. Set EIK\_HOME using this command: export EIK\_HOME = <pf\_install>/pingfederate
6. Save and close the file.
7. Enter the following command to source the bash file: . ./bash\_profile
8. Verify the new settings using the following commands, echo \$EIK\_HOME

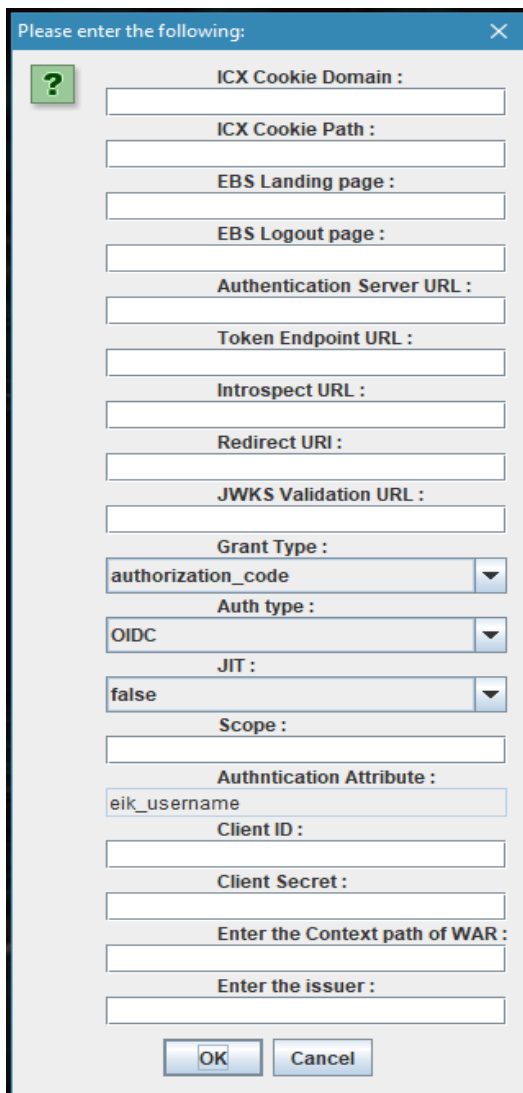
Note: If PingFederate is configured in Cluster, repeat steps 1 to 8 in all the runtime servers.

## 7.1 EIKAuth Config File Generation

1. The JAR file EIKUtility.jar needs to be placed in the \$EIK\_HOME directory created in the previous section.
2. Open terminal/command prompt (as Administrator) and navigate to EIK\_HOME directory and execute the following command

```
java -cp EIKUtility.jar com.likeminds.EBSAuth.EIKAdminUtility
```

3. On running the command, we get a dialog box as follows, Enter all values as per the configuration for all the fields using the template given below



```
#..... Configuration Panel .....#
icx_cookie_domain = <domain name> (the "." period should precede the root domain
name)
icx_cookie_path = /
ebs_landing_page = <EBS Home Page URL>
ebs_logout_page = https://<pf or pa domain name>:<port>/idp/startSLO.ping
#OIDC Configuration parameters
authentication_server_url = https://<pf/pa hostname>:<port>/as/authorization.oauth2
token_endpoint_url = https://<pf/pa hostname>:<port>/as/token.oauth2
introspect_url = https://<pf/pa hostname>:<port>/as/introspect.oauth2
redirect_uri = https://<pf/pa hostname>:<port>/EBSAuth/handler
jwks_validation_url = https://<pf/pa hostname>:<port>/pf/JWKS
grant_type = authorization_code
auth_type = OIDC
JIT = false
Scope = openid (necessary scope to return EBS username from PingFederate/LDAP store)
Authentication Attribute = eik_username (non-editable)
client_id = ***** # CLIENT_ID created on the PF console
client_secret = ***** # CLIENT_SECRET created on the PF console
Enter the context path of WAR = <Context of the war>
(i.e., For EBSAuth.war, context is /EBSAuth)
(Optional - For EBSAuthInst2.war, context is /EBSAuthInst2)
Enter the issuer = <PingFederate OAuth/OIDC issuer>
logoPath=https://likemindsconsulting.com/wp-content/uploads/2020/07/logo-1.png
helpdeskMessage=<p>If issue persists, please contact technical
<a href="https://support.likemindsconsulting.com">support</a></p>
userinfo_url=https://<pf/pa hostname>:<port>/:9031/idp/userinfo.openid
trustStorePath= https://<pf/pa hostname>:<port>/EBSAuth /cacerts
UsePrivateTrustStore=true
trustStorePassword=changeit
#..... END .....#
```

4. After entering these values in the dialog box, a file named EIKAuth.config gets generated at the EIK\_HOME directory

- Now, move the EIKAuth.config file to the following location:

<pf\_install>/pingfederate/EBSAuth

Note: If PingFederate is configured in Cluster, copy the EIKAuth.config file to all the runtime servers in the same directory location.

## 7.2 EBS DataSource (DBCX) File Creation

- Before proceeding with the DBCX file generation, a custom user should be created in the EBS Application FND\_USER table (Username – EIKUSER) with the UMX|APPS\_SCHEMA\_CONNECT role enabled.
- Place the JAR file EBSdatasource.jar in the \$EIK\_HOME directory & open the terminal/command prompt and navigate to EIK\_HOME directory to execute and generate the EBS Datasource file

```
java -cp EBSdatasource.jar com.likeminds.ebsauth.v1.EIKDatasource Y
```

Note: The jar file can also be placed & executed in any server outside of PingFederate, provided, there should be a connectivity to the EBS Database from the execution source.

- In the next step, provide the configuration details of your environment as mentioned in the screenshot:

```
C:\Users\LIKEMINDS-17\Desktop\utilities\Utilities_with_Guide>java -cp ebsdatasource-jar-with-dependencies.jar;C:\Users\LIKEMINDS-17\Desktop\utilities\Utilities_with_Guide\ojdbc8.jar;C:\Users\LIKEMINDS-17\Desktop\utilities\Utilities_with_Guide\ons.jar;C:\Users\LIKEMINDS-17\Desktop\utilities\Utilities_with_Guide\ucp.jar com.likeminds.ebsauth.v1.EIKDatasource Y
22:06:56.445 [main] INFO com.likeminds.ebsauth.v1.EIKDatasource - Name of the DBCX file with path where database configuration will be saved C:\Users\LIKEMINDS-17\Desktop\utilities\Utilities_with_Guide\eik.dbcx
Enter the connection factory class (oracle.jdbc.pool.OracleDataSource)->oracle.jdbc.pool.OracleDataSource
User entered following value ->oracle.jdbc.pool.OracleDataSource
Enter the JDBC URL:(For eg) jdbc:oracle:thin:@EBS DB domain name:<EBS DB Port>:<SID>->jdbc:oracle:thin:@<Your EBS DB domain name>:<Your EBS DB Port>:<Your SID>
User entered following value ->jdbc:oracle:thin:@<Your EBS DB domain name>:<Your EBS DB Port>:<Your SID>
Enter the apps username->apps
User entered following value ->apps
Enter the apps password->
Enter connection pool name->EBSAuth
User entered following value ->EBSAuth
Enter initial connection pool size->30
User entered following value ->30
Enter min. connection pool size->30
User entered following value ->30
Enter max. initial pool size->50
User entered following value ->50
Enter timeout check interval (Default is 30 secs)->30
User entered following value ->30
Enter inactive connection timeout (Default is 0 secs)->0
User entered following value ->0
Enter validate connection on borrow (true/false)->true
User entered following value ->true
Enter the EIK user username-><Enter your EIKUSER name>
User entered following value -><Enter your EIKUSER name>
Enter the EIK user password->
Enter the context path->/EBSAuth
User entered following value ->/EBSAuth
```

Enter the connection factory class: oracle.jdbc.pool.OracleDataSource  
Enter the JDBC URL: jdbc:oracle:thin:@<EBS DB domain name>:<EBS DB Port>:<SID>  
(Use Scan hostnames in case of RAC database)

Enter APPS username:  
Enter APPS password:  
Enter Connection pool name: EBSAuth  
Enter initial connection pool size: 10  
Enter min. connection pool size: 10  
Enter max. initial pool size: 50  
Enter timeout check interval(Default is 30 secs): 30  
Enter inactive connection timeout (Default is 0 secs): 30  
Enter validate connection on borrow (true/false): true  
Enter the EIK username: EIKUSER  
Enter the EIK user password:  
Enter the Context Path: </Context of the war file>

4. After entering all the above-mentioned values, the EIKUSER will get registered with the following message.

```
connection established
Registering User
User has been registered successfully
```

5. As part of the successful execution, a new file EIK.dbcx will be generated in the EIK\_HOME directory.
6. Now move the EIK.dbcx file to the following location:

<pf\_install>/pingfederate/EBSAuth

Note: If PingFederate is configured in Cluster, copy the EIK.DBCX file to all the runtime servers in the same directory location.

## 7.3 Deploying the log4j2.xml File

Move the provided log4j2.xml file to the following location:

```
<pf_install>/pingfederate/EBSAuth
```

Note: If PingFederate is configured in Cluster, copy the log4j2.xml file to all the runtime servers in the same directory location.

## 7.4 Deploying the cacerts File

Move the java cacerts file to the following location:

```
<pf_install>/pingfederate/EBSAuth
```

Note: If PingFederate is configured in Cluster, copy the cacerts file to all the runtime servers in the same directory location.

## 7.5 Deploying the security.properties File

Move the provided security.properties file to the following location:

```
<pf_install>/pingfederate/EBSAuth
```

Note: If PingFederate is configured in Cluster, copy the security.properties file to all the runtime servers in the same directory location.

## 7.6 Deploying EBSAuth.war File

1. To deploy the EBSAuth.war file in the PingFederate server, navigate to the “deploy” directory in the PingFederate installation directory.

```
cd /<pf_install>/pingfederate/server/default/deploy
```

2. Now copy and place the EBSAuth.war file in the above deploy directory.

Note: If PingFederate is configured in Cluster, copy the EBSAuth.war file to all the runtime servers.

3. Restart the PingFederate runtime servers (on All runtime engine nodes if it is a clustered deployment)

## 8 System Profile Parameter changes in EBS

The following Oracle EBS System Profile changes will enable SSO by delegating the authentication to PingFederate via EIK.

### 8.1 System Profile Changes

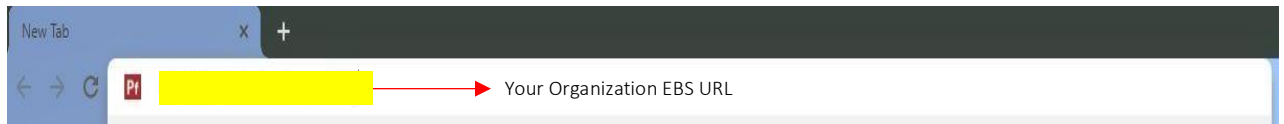
1. Login into the EBS Application using the System Admin credentials.
2. Locate and change the following EBS system profiles at Site Level for enabling SSO.

i) Application SSO Type	SSWA w/SSO
ii) Application Authenticate Agent  For Example Application server (PingFederate) PingFederate Runtime port context (case-sensitive)	<a href="https://&lt;pf-server name&gt;:&lt;pf-port&gt;/EBSAuth">https://&lt;pf-server name&gt;:&lt;pf-port&gt;/EBSAuth</a>  https://pf-server.net:9031/EBSAuth <a href="https://pf-server.net">https://pf-server.net</a> 9031 EBSAuth
iii) Application SSO Login Types	Both
iv) Application SSO Auto Link User	Enabled
v) Application SSO LDAP Synchronization	Disabled

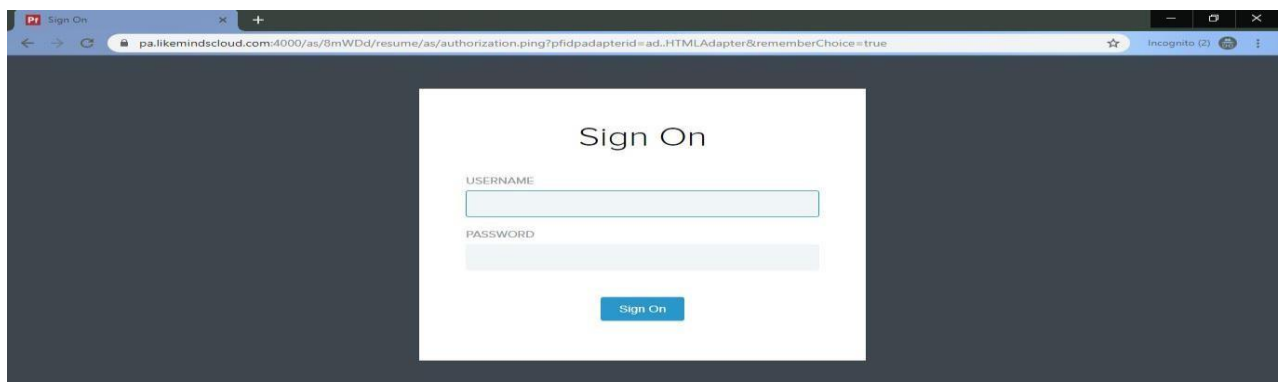
3. Restart the EBS Application Server once the changes had been made and this completes the SSO profile parameter changes in Oracle EBS.

## 9 SSO Testing for Oracle EBS

1. Enter the URL for SSO of EBS (For Example: <ebs-hostname>:<port>)



2. This URL redirects the user to Pingfederate which throws up an authentication prompt for entering the user credentials.



3. On successful validation of the user credentials, the user will get access to the EBS Homepage.



Thank you!



About LikeMinds Consulting Inc

LikeMinds consulting is a leading provider of consulting, systems integration and managed services and focuses on Identity Management, Application Security, Governance, Risk and Compliance solutions. We have focused on providing our customers with a full range of services which span through our core beliefs of Advising, Integrating, Maintaining and Accelerating the Complete Identity and Security Solution. For more information, contact us toll-free 1-888-562-3528, email [info@likemindsconsulting.com](mailto:info@likemindsconsulting.com) or visit [likemindsconsulting.com](http://likemindsconsulting.com).

© 2020 Like Minds Consulting. All rights reserved