Deploy Oracle PeopleSoft and JD Edwards Applications Across Oracle Cloud Infrastructure and Azure

SSO with Oracle Identity Cloud Service and Azure Active Directory

WHITE PAPER / JUNE 7, 2019
Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Revision History

The following revisions have been made to this white paper since its initial publication:

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 7, 2019</td>
<td>Initial publication</td>
</tr>
</tbody>
</table>

You can find the most recent versions of the Oracle Cloud Infrastructure white papers at [https://cloud.oracle.com/iaas/technical-resources](https://cloud.oracle.com/iaas/technical-resources).
Overview

When customers move Oracle PeopleSoft or JD Edwards applications to the cloud, single sign-on (SSO) for end users is still an important requirement. Most customers have some type of corporate identity and access management (IAM) solution on-premises to achieve SSO. It’s beneficial for customers to adapt a cloud SSO solution instead of trying to migrate their existing access management solution to the cloud.

This white paper provides a reference architecture and configuration details for setting up cloud SSO for PeopleSoft or JD Edwards applications running in a cross-cloud architecture, with the applications tier running on Microsoft Azure and the database tier on Oracle Cloud Infrastructure.

With this setup, a user already logged in to Azure Active Directory (Azure AD) can navigate to PeopleSoft or JD Edwards applications without logging in again, through Oracle Identity Cloud Service. Customers that deploy this solution gain the benefits of SSO, including a single set of credentials, an improved login experience, improved security, and reduced help-desk cost.

Use Cases

Following are common use cases for using SSO with PeopleSoft and JD Edwards applications and Azure AD.

- Users log in to Azure AD, access PeopleSoft or JD Edwards applications through the My Apps portal (myapps.microsoft.com portal), and get seamless access to the application.
- Application users with bookmarked PeopleSoft or JD Edwards home pages are redirected to Azure AD to log in, and get access to the application after successful authentication.
- Users who log out of PeopleSoft or JD Edwards are also logged out of the MyApps portal.

Assumptions

User synchronization between Azure AD and PeopleSoft or JD Edwards applications is a prerequisite for SSO to work. User synchronization is out of the scope of this white paper, but enterprises can use different strategies to achieve it. Typically, enterprises have an identity provisioning system that synchronizes users among Azure AD, Identity Cloud Service, and PeopleSoft or JD Edwards. Additionally, Identity Cloud Service has a feature to keep users synchronized between Azure AD and Identity Cloud Service. For more information, see the “Enabling Synchronization” section in the Identity Cloud Service documentation.
At least one attribute must match among all three systems. For example, user principal name (UPN) (or any other unique attribute) in Azure AD must match with the username or any other attribute in Identity Cloud Service, and that attribute must match with the PeopleSoft or JD Edwards application username.

This paper also assumes that the audience has hands-on experience working with the Azure platform. It doesn’t cover Azure IaaS methods or security best practices to create and run VMs and applications.

**High-Level Architecture**

Figure 1 shows a high-level architecture diagram for this SSO solution. The PeopleSoft and JD Edwards application tier is deployed on Azure and the database tier is deployed on Oracle Cloud Infrastructure. Oracle HTTP Server (OHS) acts as a reverse proxy to the application tier, which means that all the requests to the end applications go through Oracle HTTP Server. Oracle Access Manager WebGate is an Oracle HTTP Server web server plugin that intercepts every request going to the end application. If a resource being accessed is protected (requires an authenticated session), the WebGate initiates OpenID Connect (OIDC) authentication flow with Identity Cloud Service through the user’s browser. For more information about the flows supported by the OpenID Connect WebGate, see the [Oracle Access Manager documentation](#).

![SSO High-Level Architecture](#)

**Note:** For SSO configuration, it’s not relevant where the database tier is located because the WebGate never needs to make any database calls.
The WebGate (or a front-end load balancer) should be accessible to the end users and the Oracle HTTP Server host should be able to reach the Identity Cloud Service tenancy.

Identity Cloud Service redirects users to Azure AD for authentication by using the SAML 2.0 protocol. Azure AD performs the authentication, and if it is successful, the user is redirected back to the end application through Identity Cloud Service.

Figure 2 shows the end user authentication flow when an application protected with WebGate and Identity Cloud Service is accessed.

The architecture can be scaled out for high availability (HA) and failover by adding multiple Oracle HTTP Server hosts in front of an application and having a load balancer. To scale out an application deployment, follow the Azure HA and failover guidelines. See the following topics:

- **Availability Zones**
- **Load Balancer**
Configuring SSO

This section provides detailed steps for configuring SSO for PeopleSoft and JD Edwards applications and Azure AD.

Prerequisites

- An Identity Cloud Service tenancy attached to Oracle Cloud Infrastructure, and a user with the Security Administrator or Identity Domain Administrator role
- PeopleSoft and JD Edwards application administrative access accounts to change security settings
- Azure subscription with a Contributor or greater privileged account
- Azure AD subscription and a user with the Application Administrator or Global Administrator role

High-Level Steps

1. Integrate Identity Cloud Service and PeopleSoft or JD Edwards by using Oracle HTTP Server and OpenID Connect WebGate.

2. Enable PeopleSoft or JD Edwards applications for SSO.

3. Set up federation trust between Azure AD and Identity Cloud Service.

4. Create non-gallery applications in Azure AD for both PeopleSoft and JD Edwards.

Integrate Identity Cloud Service and PeopleSoft or JD Edwards by Using Oracle HTTP Server and OpenID Connect WebGate

1. Create or use an existing virtual machine (VM) in Azure.

2. Choose a certified OS for Oracle HTTP Server 12.2.1.3 installation.

3. Choose the appropriate networking components to enable Oracle HTTP Server to reach the PeopleSoft and JD Edwards application tier.

   Note: Follow the standard Azure methods and security best practices for running a VM and application installations.

4. Install Oracle HTTP Server.
5. After Oracle HTTP Server is installed, create an Oracle HTTP Server instance. Follow Azure security best practices to open Oracle HTTP Server HTTP ports.

6. Apply the latest WebGate bundle patch and deploy the WebGate to the Oracle HTTP Server instance directory.

For information about protecting an application using WebGate and Identity Cloud Service, see the following Oracle By Example article: Protecting an On-premises Application with Oracle Identity Cloud Service by Using WebGate.

**Note:** Although the article is written for protecting an on-premises application, the same set of steps apply to protecting an application deployed in the cloud.

The following steps provide sample configurations for completing each step from the Oracle By Example article, after mod_wl_ohs is configured to proxy PeopleSoft or JD Edwards applications.

7. **Register an application in Oracle Identity Cloud Service.**

A. Log in to the Oracle Identity Cloud Service console and create a trusted application. In this example, the application is named **Webgate-App**.

![Webgate-App](image)
B. Configure the application as a client.

![Webgate-App](Microsoft Azure + ORACLE)

- **Client ID:** 52e3d0f7b4f34f828ac48b6573d83347e4
- **Client Secret:** [Show Secret | Regenerate]

Note the Client ID and Client Secret values, which you will use later while configuring the WebGate.

C. Configure the client with the following values:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redirect URL</td>
<td><code>https://&lt;host:port&gt;/oauth/callback</code> &lt;br&gt;This value should match the value in the WebGate cloud.config file for the callbackPrefix parameter.</td>
</tr>
<tr>
<td>Logout URL</td>
<td><code>https://&lt;host:port&gt;/test/oauth/logout</code> &lt;br&gt;This is the URL configured for the authentication method oauth+logout in the WebGate config.policy file.</td>
</tr>
<tr>
<td>Post Logout URL</td>
<td>Oracle HTTP Server URL to the PeopleSoft or JD Edwards home page</td>
</tr>
</tbody>
</table>

![Client Configuration](Microsoft Azure + ORACLE)
D. Configure the application as a resource server. For **Secondary Audience**, enter the **host:port** of the Oracle HTTP Server instance where WebGate is configured (or, in an HA setup, enter the load balancer URL).

![Configure application APIs that need to be OAuth protected](image)

E. Activate the application.

8. **Configure WebGate to interact with Identity Cloud Service.**

Following is a sample `cloud.policy` file for protecting both PeopleSoft and JD Edwards applications with a single WebGate.

We recommend using a separate Oracle HTTP Server and WebGate pair for each application.

```json
{
    "cloudgatePolicy": {
        "comment": "Sample Cloud Policy file to protect PeopleSoft and JD Edwards application with the same WebGate",
        "disableAuthorize": false,
        "webtierPolicy": [
            {
                "policyName": "default",
                "resourceFilters": [
                    {
                        "comment": "Test Application OAuth+Logout Filter",
                        "type": "text",
                        "filter": "/test/oauth/logout",
                        "method": "oauth+logout"
                    },
                    {
                        "comment": "JDE",
                        "type": "regex",
                        "filter": "jd\d+",
                        "method": "jwt+oauth"
                    }
                ]
            }
        ]
    }
}
```
"filter" : "/jde/.*",
"method" : "oauth",
"authorize" : true,
"scope" : ",
"idcsscope" : ",
"headers" : 
	[{{"test_header" : "$subject.user.name"}}]
},
{
  "comment" : "PeopleSoft",
  "type" : "regex",
  "filter" : "/psc/.*",
  "method" : "oauth",
  "authorize" : true,
  "scope" : ",
  "idcsscope" : ",
  "headers" : 
	[{{"test_header" : "$subject.user.name"}}]
},
{
  "comment" : "PeopleSoft",
  "type" : "regex",
  "filter" : "/psp/.*",
  "method" : "oauth",
  "authorize" : true,
  "scope" : ",
  "idcsscope" : ",
  "headers" : 
	[{{"header2" : "$subject.user.name"}}]
},
{
  "comment" : "PeopleSoft",
  "type" : "regex",
  "filter" : "/ps/.*",
  "method" : "oauth",
  "authorize" : true,
  "scope" : ",
  "idcsscope" : ",
  "headers" : 
	[{{"test_header" : "$subject.user.name"}}]
},
{
  "comment" : "PeopleSoft",
  "type" : "regex",
  "filter" : "/cs/.*",
  "method" : "oauth",
  "authorize" : true,
  "scope" : ",
  "idcsscope" : ",
  "headers" : 
	[{{"test_header" : "$subject.user.name"}}]
},
{
  "comment" : "Test Application OAuth Filter",
  "type" : "regex",
  "filter" : "/cgi-bin/.*",
  "method" : "oauth",}
9. Set the headers consumed by PeopleSoft and JD Edwards.

By default, the OAM_REMOTE_USER header is set by the WebGate to the Identity Cloud Service username. The WebGate doesn’t support fetching the Identity Cloud Service username and setting it as a custom header in cloud.policy file. If an application can consume OAM_REMOTE_HEADER, no additional changes are required.

- Peoplesoft can be configured to consume the OAM_REMOTE_USER header set by the WebGate. If a different header name is used (for example, PSUSER), add the following line to the Oracle HTTP Server instance httpd.conf file:
  
  \[RequestHeader set PSUSER "\{OAM_REMOTE_USER\}e"

  Save, and then restart the Oracle HTTP Server instance.

- JD Edwards expects the WebGate to set the JDE_SSO_UID header. It can’t be configured to read the OAM_REMOTE_USER header. To set the JDE_SSO_UID header, add the following line to the Oracle HTTP Server instance httpd.conf file:
  
  \[RequestHeader set JDE_SSO_UID "\{OAM_REMOTE_USER\}e"

  Save, and then restart the Oracle HTTP Server instance.

Enable PeopleSoft or JD Edwards Applications for SSO

Identity Cloud Service integrates with PeopleSoft and JD Edwards deployed on-premises or in Oracle Cloud Infrastructure by using a component called App Gateway. Because App Gateway isn’t supported on the Azure platform, this integration is done here through Oracle HTTP Server and WebGate. The following links go to the App Gateway documentation, but the steps to enable SSO for the end applications using Oracle HTTP Server and WebGate are the same.

1. Enable SSO for PeopleSoft.

2. Enable SSO for JD Edwards.

3. Create a test user in PeopleSoft or JD Edwards and Identity Cloud Service so that the Identity Cloud Service username matches the PeopleSoft or JD Edwards username.
4. Verify the integration by accessing the PeopleSoft or JD Edwards home page.
   You should be redirected to Identity Cloud Service for login. After successful authentication, the test user should be logged in to the end application.

Set Up Federation Trust Between Azure AD and Identity Cloud Service

This step has several parts:

1. Add Identity Cloud Service as a gallery application in Azure AD.
2. Add Azure AD as an identity provider (IDP) in Identity Cloud Service.
3. Complete single sign-on configuration in Azure AD.

Add Identity Cloud Service as a Gallery Application in Azure AD

1. Log in to your Identity Cloud Service (IDCS) tenancy with admin credentials, go to your tenancy-specific metadata URL, and download the metadata. The URL looks as follows:

   https://<your_tenancy>.identity.oraclecloud.com/fed/v1/metadata

2. In the Azure portal, navigate to Azure Active Directory > Enterprise applications > All applications, and then click New application.

3. Depending on which application you are protecting, type Oracle IDCS for PeopleSoft or Oracle IDCS for JD Edwards in the search box, and select the matching application from the resulting applications.
4. For the new application, navigate to **Single sign-on** and select **SAML** as the single sign-on method.

5. Click **Upload metadata file**.

6. Select the Identity Cloud Service metadata file that you downloaded in step A, and then click **Add**.
7. Verify the SAML configuration. Add the Oracle Identity Cloud Service Logout Url, if it’s missing. In the User Attributes & Claims section, keep the default values.

8. In the SAML Signing Certificate section, click Download next to Federation Metadata XML to download the Azure AD federation metadata file.

This application provides a SAML 2.0 federation link between Azure AD and Identity Cloud Service, but PeopleSoft and JD Edwards application users should see only the PeopleSoft or JD Edwards application in the My Apps portal.
9. To hide the application in the My Apps portal, set the **Visible to users?** property to **No**.

Add Azure AD as an IDP in Identity Cloud Service

1. Log in to the Identity Cloud Service admin console.

2. In the navigation pane, click **Security**, click **Identity Providers**, and then add an IDP.

3. In the wizard, enter a name for the IDP, and then click **Next**.
4. Import the Azure AD Federation Metadata file that you downloaded in the previous section.

5. Use the default value for **Requested NameID Format**. The value for **Identity Provider User Attribute** should be **Name ID**. Set the value for **Oracle Identity Cloud Service User Attribute** to **Primary Email Address** or to any other attribute in Identity Cloud Service that might hold the user principal name in Azure AD.

6. Click **Save**.
   
   For more information about adding an IDP to Identity Cloud Service, see the [documentation](#).
7. Set up an IDP policy and add each app that might use Azure AD for authentication.

A. In the navigation pane, click **Security**, and then click **IDP Policies**.

B. Click **Add**.

C. In the wizard, enter the name for the policy, and then click **Next**.

D. Click **Assign**, select **Azure AD IDP** from the list, and then click **Next**.

E. Assign one or more applications that might use this IDP.

Complete Single Sign-On Configuration in Azure AD

1. Sign in to the [Azure portal](https://azure.microsoft.com).

2. Create a security group, for example, **JDE-Users**.
3. **Create a test user.**

4. **Add the user to the group.**

5. **Assign the group to the IDCS SSO application.**

   For example, the JDE-Users group contains all the users who might access a JD Edwards application through Identity Cloud Service.
6. Open the Identity Cloud Service admin console.

For testing purposes, you can either create a user in Identity Cloud Service manually or synchronize Azure AD users in Identity Cloud Service.

The users should be created or synchronized such that a user principal name in Azure AD matches the user’s primary email address (or some other attribute) in Identity Cloud Service. For example, joe.smith@example.com would be the user’s principal name in Azure AD and the Identity Cloud Service primary email address.

7. In Azure AD, navigate to the IDCS-SSO enterprise application and test single-sign on by using the test account.
Create a Non-Gallery Application for PeopleSoft or JD Edwards

1. In the Azure portal, navigate to Azure Active Directory > Enterprise applications > All applications, and then click New application.

2. Click Non-gallery application.

3. Enter a name (for example, PeopleSoft or JDE) in the Add your own application pane, and then click Add.

4. For the new application, navigate to Single sign-on and select Linked as the single sign-on method.
5. Enter the URL of the PeopleSoft or JD Edwards application home page, and click **Save**.

6. In the properties for the application, verify that **User assignment required?** and **Visible to users?** are set to **Yes**.

7. **Create users, security groups, and group memberships in Azure.**

8. Assign group-based access to each application in Azure AD. You can create **PeopleSoft-Users** and **JDE-Users** groups and assign **PeopleSoft-Users** to the **PeopleSoft** application and **JDE-Users** to the **JDE** application.

   For more information about configuring single sign-on to non-gallery apps in Microsoft, see the [Microsoft documentation](#).
9. After the JD Edwards or PeopleSoft application is configured in Azure AD, you can configure [conditional access policies in Azure AD](https://aad.microsoft.com) to enhance sign-on security.

Verifying SSO

These steps show verifying JD Edwards, but you would follow the same set of steps for verifying PeopleSoft application SSO.

1. Ensure that a test user is present in Azure AD, Identity Cloud Service, and PeopleSoft or JD Edwards and is assigned the right group.

The user principal name (UPN) in Azure AD should match the Identity Cloud Service email or username.
The Identity Cloud Service username should match the PeopleSoft or JD Edwards username.

2. Log in to the My Apps portal with the test user credentials.

3. Click the application (for example, JDE).
   The user should be logged in to the application and sees the home page.
4. View **My System Profile** to verify the logged in username.

![Image of Oracle JD Edwards interface]

**Conclusion**

This paper describes how to set up Azure AD as the IDP for Oracle applications deployed on a cross-cloud platform. It covers a pattern for setting up SSO with Oracle applications like PeopleSoft and JD Edwards in which Identity Cloud Service acts as a bridge between the applications and Azure AD. This setup enables scenarios in which customers can host Oracle Databases in Oracle Cloud Infrastructure while using Azure AD as their identity provider.