# Table of Contents

1. WSO2 Identity Cloud Documentation .................................................. 3  
   1.1 Introduction .................................................................................. 3  
      1.1.1 Overview .............................................................................. 4  
      1.1.2 Architecture ......................................................................... 4  
   1.2 Quick Start Guide ........................................................................ 8  
   1.3 Key Concepts ............................................................................... 14  
   1.4 Tutorials .................................................................................... 23  
      1.4.1 Configuring an On-premise User Store .................................. 24  
      1.4.2 Configuring SSO for Custom Applications ........................... 28  
         1.4.2.1 Configuring Standards-based SSO ................................. 30  
            1.4.2.1.1 Configuring SAML SSO ....................................... 31  
            1.4.2.1.2 Configuring OpenID Connect SSO ......................... 42  
            1.4.2.1.3 Configuring WS-Federation SSO ......................... 46  
         1.4.2.2 Configuring Proxy-based SSO ........................................ 49  
            1.4.2.3 Configuring a Shortcut type Application ...................... 52  
      1.4.3 Configuring SSO for Third-party Applications ....................... 55  
         1.4.3.1 Configuring SSO for Salesforce ..................................... 55  
         1.4.3.2 Configuring SSO for NetSuite ......................................... 71  
         1.4.3.3 Configuring SSO for Zuora ............................................ 79  
         1.4.3.4 Configuring SSO for AWS ........................................... 79  
         1.4.3.5 Configuring SSO for GoToMeeting ................................. 90  
         1.4.3.6 Configuring SSO for Concur .......................................... 95  
      1.4.4 Configuring Custom Themes .................................................. 99  
      1.4.5 Configuring Custom User Portal URL .................................... 101  
   1.5 Cloud Administration ..................................................................... 107  
   1.6 FAQ ............................................................................................ 115
WSO2 Identity Cloud delvers enterprise-ready solution for identity management and setting up single sign-on for multiple applications.

<table>
<thead>
<tr>
<th>Get started with WSO2 Identity Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are new to using WSO2 Identity Cloud, follow the steps given below to get started:</td>
</tr>
<tr>
<td><strong>Introduction</strong></td>
</tr>
<tr>
<td><strong>Quick Start Guide</strong></td>
</tr>
<tr>
<td><strong>Tutorials</strong></td>
</tr>
</tbody>
</table>

To download a PDF of this document or a selected part of it, click [here](#) (only generate one PDF at a time). You can also use this link to export to HTML or XML.
Introduction

This section covers the following:

- Overview
- Architecture

Overview

WSO2 Identity Cloud is an enterprise-ready service that enables you to set up your applications for single sign-on and connect them to your LDAP user store. The Identity Cloud is based on the WSO2 Identity Server, WSO2's complete, on-premise solution for managing identity and access.

WSO2 Identity Cloud provides extensible support to easily configure single sign-on for enterprise and popular SaaS applications for the users in an organization's internal user store (LDAP). The organization's local LDAP can be configured as an on-premise user store and access is provided to the Identity Cloud without sharing LDAP credentials with the Identity Cloud.

Identity Cloud provides two separate login portals with separate URLs for each portal. These portals are used by two different types of users.

- The organization's admins - The Admin Portal allows you to configure application security for authentication standards, such as SAML, OIDC, and WS-Federation.
- Normal users (employees of the organization) - The User Portal provides a central location for the organization's users to log in and discover applications in a central place, while applications can be accessed with single sign-on.

Architecture

WSO2 Identity Cloud is deployed on top of WSO2 Identity Server. The following diagram depicts the overall architecture of WSO2 Identity Cloud.

Figure: WSO2 Identity Cloud architecture
The following sections expand on the above architecture diagram.

- **Architecture components**
- **Architecture and the business scenario of the Identity Cloud**

## Architecture components

The WSO2 Identity Cloud architecture consists of the following key components.

- **Admin Portal**
- **User Portal**
- **Identity Provider**
- **Identity Gateway**
- **User Store Agent**

These are explained as follows.

**Admin Portal**

The *Admin Portal* is the place where organization admins can log in and configure security and single sign-on for their applications with authentication standards, such as SAML, OpenID Connect, and WS-Federation. Applications configured here are listed in **User Portal**, so that other users in the organization can discover the applications. In addition to configuring security for applications, admin portal allows defining which users can view which applications in **User Portal** depending on the user roles. There are multiple options provided in simplified UI to configure security for applications depending on your application nature and how you want to handle security. Following is the list of options provided to configure SSO.

<table>
<thead>
<tr>
<th>Option to configure SSO</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Predefined Applications</strong></td>
<td>The WSO2 Identity Cloud has a selected set of popular cloud applications and has identified the minimum configurations that a user must provide to configure SSO for these applications. The configurations differ from application to application depending on the service provider. This is done to simplify the process for users so that they need to provide only the minimum configurations necessary. Currently, the Identity Cloud has identified and supports Salesforce, Concur, AWS, NetSuite, Zuora, and GotoMeeting applications as predefined applications. This list will keep growing in future. If you can see the application that you are going to configure SSO for under the predefined application list, it makes the process easier for you to select the application from there and provide the required configuration details. If the application that you are trying to configure is not listed in predefined application list, you can configure it as a “Custom Application”.</td>
</tr>
<tr>
<td><strong>Custom Applications</strong></td>
<td>This option can be used when the application that you need to configure is not listed in the predefined applications list or if there are any additional configurations that you need to do other than the minimum set of configurations provided in the predefined applications. Custom applications are divided into three types based on how SSO is handled in the application. These are listed out in the following sections of this table.</td>
</tr>
<tr>
<td><strong>Agent Type Applications</strong></td>
<td>These applications must handle the SSO logic. For example, when an application initiates SSO (using a SAML request) and processes the SAML response sent by the identity provider.</td>
</tr>
<tr>
<td><strong>Proxy Type Applications</strong></td>
<td>Proxy-type applications do not need to handle the SSO logic. There is a central gateway provided that acts as a proxy for the application, handles SSO requests/responses on behalf of the application, and sends a signed JWT token to the application where it can process it and identify the authenticated user.</td>
</tr>
</tbody>
</table>
Shortcut Type  | This is the only application listing in the user portal. This type can be used when you have websites or applications having their own authentication. However, these applications must be listed in a central place along with all the other applications.

**User Portal**

The *User Portal* provides a central location for the users of an organization to log in and discover applications in a central place, while applications can be accessed with single sign-on.

**Identity Provider**

The *Identity Provider* is the key component that handles all security related operations.

**Identity Gateway**

The *Identity Gateway* is a runtime, backend component developed using the WSO2 ESB. The *Identity Gateway* is a simple application proxy that intercepts application requests and applies policies such as throttling and security checks.

**User Store Agent**

The *User Store Agent* is a microservice developed with WSO2 MSF4J to connect the on-premise LDAP of an organization to the WSO2 Identity Cloud. This allows organizations to provide access to applications with single sign-on, provided that their credentials are stored in the organization’s LDAP. This can be done without sharing the credentials of LDAP with the WSO2 Identity Cloud.

**Architecture and the business scenario of the Identity Cloud**

The following table expands on the business scenario of the WSO2 Identity Cloud and how this involves the architecture.

<table>
<thead>
<tr>
<th>Process</th>
<th>Description of how this works from an architectural perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring on-premise user store</td>
<td>The <em>User Store Agent</em> must be downloaded to an on-premise location and configured to connect to your local LDAP server. This is represented by the following section of the architecture diagram.</td>
</tr>
</tbody>
</table>

*Figure: On-premise User Store Agent and LDAP server*

The identities of the users are stored in the LDAP and these credentials are used to access the WSO2 Identity Cloud and the various other applications that the user needs to interact with.
Creating applications

In the architecture diagram, the integral part of the flow involves the admin user creating applications in the Admin Portal. These are saved into the database in the WSO2 Identity Cloud.

*Figure*: Admin user creating apps using the Admin Portal in the Identity Cloud

These can be any custom application or a common SaaS application. The Identity Cloud can support applications that use SAML 2.0, OpenID Connect, and WS-Federation protocols.

Accessing applications using SSO

The user can access the applications created in the Admin Portal through the User Portal, which obtains the configured application list from the database. More significantly, the users can access these applications in a single sign-on manner.

*Figure*: User views the applications in the User Portal and accesses them using SSO
The WSO2 Identity Cloud is able to handle different authentication requests.

There are three possible ways an authentication request can come to the identity provider from the client.

- **Service provider initialized SSO** - The client invokes the application URL. The application has the logic to initiate a single sign-on request to the identity provider.
- **Identity provider initialized SSO** - The client invokes a URL given by the identity provider instead of the web application URL. So, the identity provider itself initializes the single sign-on request on behalf of the application.
- **Proxy application** - The client invokes a proxy URL given by the Identity Gateway instead of a real web application URL. The proxy (gateway) initializes the single sign-on request to the identity provider on behalf of application.

This is depicted in the following section of the architecture.

*Figure: Authentication request handling*
Quick Start Guide

The WSO2 Identity Cloud uses the identities of the users in your local LDAP user store to provide single sign-on (SSO) access to multitudes of Software-as-a-Service (SaaS) applications. A representation of these applications can be added into the WSO2 Identity Cloud and are displayed in an application catalogue.

![Diagram: Adding applications and configuring them for SSO with an on-premise user store]

**Figure: Adding applications and configuring them for SSO with an on-premise user store**

Let's look at the basic business use cases of the WSO2 Identity Cloud and illustrate them using samples.

- Introducing the samples
- Accessing Identity Cloud and deploying the samples
- Accessing the sample applications using SSO
- What's next?

Introducing the samples

The WSO2 Identity Cloud comes with sample data for you to try out features like SSO. Sample data consists of two parts, a sample user store and few sample applications.

Sample user store

The main feature of the Identity Cloud is providing single sign-on (SSO) for the applications in an organization, so that the users of the organization can use all the applications without signing in to each and every application. WSO2 Identity Cloud provides you with a sample user store to try out single sign-on. Once you are done with the samples, you can connect the user store of your organization to the WSO2 Identity Cloud.

Sample applications

Sample applications are there to demonstrate how single sign-on works for your applications. These applications are similar to any third-party applications that have separate independent session management from the WSO2 Identity Cloud. We have used WSO2 Identity Cloud to illustrate single sign-on with these sample applications.

**Note:** The sample applications use SAML 2.0 protocol. Applications that use OpenID Connect and WS-Federation can also be configured.

Accessing Identity Cloud and deploying the samples
The sample data is not deployed by default. Once you create an organization and log in to the admin portal using your cloud credentials, the sample data can be deployed.

1. If you have not registered to the WSO2 Identity Cloud yet, go to [https://cloud.wso2.com/](https://cloud.wso2.com/) and sign up.

2. When you log into the WSO2 Identity Cloud for the first time, you will see following screen. Click **Start with sample data** to configure a sample user store and sample applications. It takes a few seconds to deploy the sample user store and sample applications.

3. In the page that appears, you can see a notification that states that you are using a sample user store now. You can also see a list of sample users. Click the icon next to the password to see the password for a user. Make a note of the credentials of the user. Click **Try out Sample Applications**.

---

**Tip**: Clicking the **Start with sample data** link ensures that you have a sample user store configured to the WSO2 Identity Cloud. It also ascertains that two sample applications are added to the catalogue of applications associated with your account.
Accessing the sample applications using SSO

The next step is to access the sample applications in the WSO2 Identity Cloud using single sign-on.

1. Once you click the **Try out Sample Applications** button, a new window opens and you can see the catalogue of applications for your account.

2. Right-click on an application and open it in a new private window or copy the link into a different browser. Click the **LOGIN** button.
3. You are requested to log into the WSO2 Cloud. Use the credentials that were generated for the sample user store and click **Sign In**.

4. You are now logged into this application.
5. Now open the other application in the same browser that you opened the first application. You are automatically logged into this application without having to enter any credentials.

You have now configured single sign-on using two sample applications and a sample user store.

What's next?
Now that you have set up sample applications and a sample user store, it is time to try this out using actual applications and actual user stores. Also you can learn more about the concepts associated with single sign-on.

- You can use your own LDAP and add your own applications to set up single sign-on for the applications in your enterprise. To learn how to do this, see the Tutorials.
- If you need more information on the concept of single sign-on and the protocols that can be used, see Key Concepts.
Key Concepts

This section includes the following key concepts that are pertaining to the WSO2 Identity Cloud.

Introducing single sign-on

Single sign-on (SSO) is one of the key features of the WSO2 Identity Cloud that enables users to provide their credentials once and obtain access to multiple applications. The users are not prompted for their credentials when accessing each application until their session is terminated. Additionally, the user can access all these applications without having to log into each and every one of them individually. So, if users log into application A, for example, they would automatically have access to application B as well for the duration of that session without having to re-enter their credentials.

The following are some of the advantages you can have with SSO:

- Users need only a single username/password pair to access multiple services. Thus they do not have the issue of remembering multiple username/password pairs.
- Users are authenticated only once at the identity provider and then they are automatically logged into all services within that “trust-domain”. This process is more convenient to users since they do not have to provide their username/password at every service provider.
- Service providers do not have the overhead of managing user identities, which is more convenient for them.
- User identities are managed at a central point. This is more secure, less complex and easily manageable.

How single sign-on works

To understand how single sign-on works, it is useful to first examine how authentication works in a non SSO environment. The following diagram depicts this flow.

Figure: Authentication in a non SSO environment
In the above figure, the users access Application 1 first and then Application 2 using the same browser. The user is authenticated in Application 1 first and can access this application. When the users attempt to access Application 2, they must enter their credentials again and get access to Application 2 separately. There is no way for Application 2 to access the cookie stored in the browser for Application 1 and authenticate the user by that means.

The following diagram depicts how this story differs in a SSO environment.

![SSO Diagram](image)

**Figure: Authentication in a SSO environment**

In this second figure, the users access Application 1 first followed by Application 2. Once again, the users use the same browser to access these applications. The users are redirected to the WSO2 Identity Cloud (or any authentication server that is configured for this task). If the users are already logged in to the Identity Cloud, the user is authenticated in Application 1 via an authentication token sent by the Identity Cloud to Application 1. When Application 2 is accessed, the same thing happens and the users are redirected to the Identity Cloud and authenticated. This is how SSO works.

**SSO in reality**

Single sign-on is widely used in web technologies. Google is one of the best examples. Try this simple exercise.

1. Visit [www.google.com](http://www.google.com) from your web browser.
2. Click the **SIGN IN** button on the top right of the page.
3. Once you sign in, you are redirected to [https://accounts.google.com/ServiceLogin](https://accounts.google.com/ServiceLogin). There you are requested to enter your Username and Password. Enter your Google credentials there.
4. Once you enter your Username and Password, you are directed back to [www.google.com](http://www.google.com) where you started.
5. Next visit [www.gmail.com](http://www.gmail.com), the Google mail server.
6. Notice that you are automatically signed in and you directly access your Gmail Inbox. You did not have to enter your Username and Password at Gmail.
7. In addition to that; now try [www.youtube.com](http://www.youtube.com).
8. You are automatically signed in. You do not have to enter your username and password at YouTube.
Single Sign-On (SSO) allows you to sign in only once but provides access to multiple resources without having to re-enter your username and password.

Supported protocols for agent-based SSO

The following are the supported protocols for agent-based single sign-on.

- SAML 2.0
- WS-Federation
- OpenID Connect

The following sections in this topic expand on these protocols and provide details pertaining to them.

**Recommended reading**

See Configuring Standards-based SSO for more information on how to configure each of these protocols for agent-based SSO in the WSO2 Identity Cloud.

**SAML 2.0 SSO**

SAML stands for Security Assertion Markup Language, which is an XML-based data format for exchanging authentication and authorization data between an identity provider and a service provider. The single most important requirement that SAML addresses is web browser single sign-on (SSO). Three main roles are defined in SAML Specification.

- **The Principal**: This is typically the user who requires a service from a service provider entity.
- **The Identity Provider**: The SAML authority that provides the identity assertion to authenticate a principal.
- **The Service Provider**: The SAML consumer that provides the service for principals.

The main use case scenario covered by SAML is the principal (the user) requesting access to a resource or service from the service provider. Then the service provider, using SAML, communicates with the identity provider to obtain identity assertion. The service provider makes the access control decision, depending on this assertion.

SAML 2.0 is the latest version of SAML, which uses security tokens containing assertions to pass information about a user between an identity provider and a service provider.

SAML 2.0 provides five main specifications:

- Core
- Bindings
- Profiles
- Metadata
- Conformances

**SAML 2.0 web browser-based SSO profile**

SAML 2.0 Web Browser based SSO profile is defined under the SAML 2.0 Profiles specification.

In a web browser based SSO system, the flow can be started by the user either by attempting to access a service at the service provider or by directly accessing the identity provider itself.

If the user accesses a service at a service provider:

**Tip**: Notice the URL of the web browser. Each time you access an application, you see that you are being redirected to https://accounts.google.com/ServiceLogin and return immediately back to the website.
1. The service provider determines which identity provider to use (this is the case when there are multiple identity providers. SAML identity provider discovery profile may be used).
2. The service provider generates a SAML message and then redirects the web browser to the identity provider along with the message.
3. Identity provider authenticates the user.
4. The identity provider generates a SAML message and then redirects the web browser back to the service provider.
5. The service provider processes the SAML message and decides to grant or deny access to the user.

If the user accesses the identity provider directly, then only the steps 3, 4 and 5 are in the flow.

The following diagram depicts this flow.

**Figure: SAML 2.0 authentication flow**

The message MUST contain an element that uniquely identifies the service provider who created the message. Optionally the message may contain elements such as Issuer, NameIDPolicy, etc. More information regarding the message can be found in SAML Core Specification.

Service providers act as SAML assertion consumers. They have two basic functions:

- Create messages and redirect users to the identity provider with the created message.
- Process messages from the identity provider and make decisions based on them.

The following code is a sketch of a sample service provider servlet in a SAML 2.0 web browser-based SSO system.
public class Resource extends HttpServlet
{
    private static SamlConsumer consumer = new SamlConsumer();

    public void doGet(HttpServletRequest request, HttpServletResponse response)
    {
        requestMessage = consumer.buildRequestMessage();
        response.sendRedirect(requestMessage);
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
    {
        responseMessage = request.getParameter("SAMLResponse").toString();
        result = consumer.processResponseMessage(responseMessage);
    }
}

When a web user attempts to access the above servlet, its `doGet()` method is called. Inside the `doGet()` method, it generates a message and then redirects the user to the identity provider.

After authentication is completed by the identity provider, it does a POST callback to the above servlet with a message. Then the `doPost()` method of the servlet gets called and inside the `doPost()` method, it retrieves the message from the request and then the message is passed to the `SamlConsumer` instance for processing.

<AuthnRequest> message

To create an `<AuthnRequest>` message using the OpenSAML library, do the following.

1. Add the OpenSAML library to the build path of the project. You can download the OpenSAML .jar file from here.
2. A sample `<AuthnRequest>` message can be found here.
3. According to SAML 2.0 specifications, the message must contain an element. Create the Issuer element first.

```java
String issuerId = "saml2.sso.demo";
IssuerBuilder issuerBuilder = new IssuerBuilder();
Issuer issuer = issuerBuilder.buildObject("urn:oasis:names:tc:SAML:2.0:assertion", "Issuer", "samlp");
issuer.setValue(issuerId);
```
4. Create the `<AuthnRequest>` next.

```java
// the issuerUrl is the url of the service provider who generates the message
String issuerUrl =
    "http://localhost:8080/saml2.sso.demo/consumer";
DateTime issueInstant = new DateTime();
AuthnRequestBuilder authnRequestBuilder = new AuthnRequestBuilder();
AuthnRequest authnRequest =
    authnRequestBuilder.buildObject("urn:oasis:names:tc:SAML:2.0:protocol", "AuthnRequest", "samlp");
authnRequest.setForceAuthn(false);
authnRequest.setIsPassive(false);
authnRequest.setIssueInstant(issueInstant);
authnRequest.setAssertionConsumerServiceURL(issuerUrl);
authnRequest.setIssuer(issuer);
authnRequest.setID(aRandomId);
authnRequest.setVersion(SAMLVersion.VERSION_20);
```

The message may contain many other elements like ",," etc. those elements can be created and added to the message in the same way.

5. Next encode the message.

```java
Unmarshaller marshaller =
    Configuration.getMarshallerFactory().getMarshaller(authnRequest);
Element authDOM = marshaller.marshal(authnRequest);

StringWriter rspWrt = new StringWriter();
XMLHelper.writeNode(authDOM, rspWrt);
String requestMessage = rspWrt.toString();

Deflater deflater = new Deflater(Deflater.DEFLATED, true);
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
DeflaterOutputStream deflaterOutputStream = new DeflaterOutputStream(byteArrayOutputStream, deflater);
deflaterOutputStream.write(requestMessage.getBytes());
deflaterOutputStream.close();
/* Encoding the compressed message */
String encodedRequestMessage =
    Base64.encodeBytes(byteArrayOutputStream.toByteArray(), Base64.DONT_BREAK_LINES);
String encodedAuthnRequest =
    URLEncoder.encode(encodedRequestMessage, "UTF-8").trim();
```
6. Construct the redirection URL.

```java
redirectionUrl = identityProviderUrl + "?SAMLRequest=" + encodedRequestMessage;
```

7. Redirect the user to the identity provider.

```java
response.sendRedirect(redirectionUrl);
```

<Response> message

To read the <Response> message issued by the WSO2 Identity Cloud:
1. A sample `<Response>` message can be found here.
2. The response message must be fetched from the request.

   ```java
   responseMessage = request.getParameter("SAMLResponse").toString();
   ```
3. The fetched `responseMessage` is unmarshaled and the SAML message is retrieved.

   ```java
   documentBuilderFactory.setNamespaceAware(true);
   DocumentBuilder docBuilder = documentBuilderFactory.newDocumentBuilder();
   byte[] base64DecodedResponse = Base64.decode(responseMessage);
   ByteArrayInputStream is = new ByteArrayInputStream(base64DecodedResponse);
   Document document = docBuilder.parse(is);
   Element element = document.getDocumentElement();
   UnmarshallerFactory unmarshallerFactory = Configuration.getUnmarshallerFactory();
   Unmarshaller unmarshaller = unmarshallerFactory.getUnmarshaller(element);
   Response response = (Response) unmarshaller.unmarshal(element);
   ```
4. The retrieved SAML 2.0 response message can be easily processed. For example, let's take the username or the subject's NameID.

   ```java
   String subject = response.getAssertions().get(0).getSubject().getNameID().getValue();
   ```
5. Alternatively, if the response is signed by the identity provider, you can retrieve the certificate.

   ```java
   String certificate = response.getSignature().getKeyInfo().getX509Datas().get(0).getX509Certificates().get(0).getValue();
   ```

**Recommended reading**

See SAML 2.0 specification documentation for more information.
**WS-Federation**

WS-Federation (Web Services Federation) describes the management and brokering of trust relationships and security token exchange across Web services and organizational boundaries. WS-Federation is a part of the larger WS-Security framework. For example, WS-Federation builds on the Security Token Service (STS) by providing mechanisms that facilitate interactions. In the WS-Federation Model, an Identity Provider is a Security Token Service (STS).

<table>
<thead>
<tr>
<th>Recommended reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>See WS-Federation specification documentation for more information.</td>
</tr>
</tbody>
</table>

**OpenID Connect**

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OpenID Connect allows clients of all types, including Web-based, mobile, and JavaScript clients, to request and receive information about authenticated sessions and end-users. The specification suite is extensible, allowing participants to use optional features such as encryption of identity data, discovery of OpenID Providers, and session management, when it makes sense for them.

<table>
<thead>
<tr>
<th>Recommended reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>See OpenID Connect specification documentation for more information.</td>
</tr>
</tbody>
</table>
Tutorials

The following are the tutorials available for the WSO2 Identity Cloud. They can be considered as standalone tutorials so navigate to the tutorials that are most relevant to your scenario.

- Configuring an On-premise User Store
- Configuring SSO for Custom Applications
  - Configuring Standards-based SSO
    - Configuring SAML SSO
    - Configuring OpenID Connect SSO
    - Configuring WS-Federation SSO
  - Configuring Proxy-based SSO
  - Configuring a Shortcut type Application
- Configuring SSO for Third-party Applications
  - Configuring SSO for Salesforce
  - Configuring SSO for NetSuite
  - Configuring SSO for Zuora
  - Configuring SSO for AWS
  - Configuring SSO for GoToMeeting
  - Configuring SSO for Concur
- Configuring Custom Themes
- Configuring Custom User Portal URL

Configuring an On-premise User Store

WSO2 Identity Cloud provides single sign-on (SSO) capabilities for the applications in an organization so that the users of the organization can use all the applications seamlessly without having to sign in to each and every application separately. You can connect the on-premise user store of the organization directly to WSO2 Identity Cloud to enable this.

An outbound agent is used to connect the organization's local LDAP to WSO2 Identity Cloud. This allows the organization to give application access (with SSO) for users in the LDAP, without sharing the credentials of the LDAP with WSO2 Identity Cloud.

**In this tutorial you will learn** how to connect an on-premise user store of your organization to WSO2 Identity Cloud.

**Important!** Java 1.8 or a later version is required to run the agent. Ensure that the correct Java version is installed in your server.

1. Sign up and log into WSO2 Identity Cloud.
2. Click on the menu bar on the top left corner.
3. Click **Directories** from the left menu.

This takes a few seconds to complete and it begins downloading the agent file. This performs some backend operations in Identity Cloud that is required to connect to the on-premise user store and you are redirected to the following screen.

4. The following screen is displayed. Click the **Connect my LDAP to Cloud** button.

   ![Identity Cloud Screen](image)

   **Note**: If you are unable to download the agent, click **DOWNLOAD AGENT** to explicitly download the agent.
5. Unzip the downloaded agent file. Open the `<AGENT_HOME>/conf/userstore-config.xml` file and do the required changes to point to your LDAP (or any other LDAP you require access to).

   - Click to view a sample userstore-config.xml file

   ```xml
   <UserStoreManager class="org.wso2.carbon.identity.agent.userstore.manager.ldap.LDAPUserStoreManager">
     <Property name="ConnectionURL">ldap://localhost:10389</Property>
     <Property name="ConnectionName">uid=admin,ou=system</Property>
     <Property name="ConnectionPassword">admin</Property>
     <Property name="UserSearchBase">ou=system</Property>
     <Property name="UserNameAttribute">uid</Property>
     <Property name="UserNameSearchFilter">(&amp;(objectClass=person)(uid=?))</Property>
     <Property name="UserNameListFilter">(objectClass=person)</Property>
     <Property name="GroupNameAttribute">cn</Property>
     <Property name="GroupSearchBase">ou=system</Property>
     <Property name="GroupNameListFilter">(objectClass=groupOfNames)</Property>
     <Property name="GroupNameSearchFilter">(&(objectClass=groupOfNames)(cn=?))</Property>
     <Property name="MembershipAttribute">member</Property>
     <Property name="EmptyRolesAllowed">true</Property>
   </UserStoreManager>
   ```

   - Click to view descriptions of the key properties you use to configure the on-premise user stores

   The following table provides descriptions of the key properties in the `userstore-config.xml` file you use to configure on-premise user stores.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionURL</td>
<td>Connection URL to the user store server. In the case of default LDAP in Carbon, the port is specified in the <code>carbon.xml</code> file, and a reference to that port is included in this configuration.</td>
</tr>
<tr>
<td>ConnectionName</td>
<td>The username used to connect to the database and perform various operations. This user does not have to be an administrator in the user store or have an administrator role in the WSO2 product that you are using, but this user MUST have permissions to read the user list and users' attributes and to perform search operations on the user store. The value you specify is used as the DN (Distinguish Name) attribute of the user. This property is mandatory.</td>
</tr>
<tr>
<td>ConnectionPassword</td>
<td>Password for the <code>ConnectionName</code> user.</td>
</tr>
</tbody>
</table>
UserNameListFilter | Filtering criteria for listing all the user entries in the user store. This query or filter is used when doing search operations on users. In this case, the search operation only provides the objects created from the specified class. This query is the same as listing out all the available users in the management console.

UserSearchBase | DN of the context or object under which the user entries are stored in the user store. In this case, it is the "users" container. When the user store searches for users, it will start from this location of the directory.

UserNameSearchFilter | Filtering criteria used to search for a particular user entry.

UserNameAttribute | The attribute used for uniquely identifying a user entry. Users can be authenticated using their email address, UID, etc.

EmptyRolesAllowed | Specifies whether the underlying user store allows empty groups to be created. In the case of LDAP in Carbon, the schema is modified such that empty groups are allowed to be created. Usually LDAP servers do not allow you to create empty groups.

GroupSearchBase | DN of the context under which user entries are stored in the user store.

GroupNameListFilter | Filtering criteria for listing all the group entries in the user store. Groups are created in LDAP using the "groupOfName" class. The group search operation only returns objects created from this class.

GroupNameSearchFilter | Filtering criteria used to search for a particular group entry.

GroupNameAttribute | Attribute used for uniquely identifying a user entry. This attribute is to be treated as the group name.

MembershipAttribute | Attribute used to define members of groups.

6. To start the agent, run the script wso2agent.sh (on Linux/Mac OS) or wso2agent.bat (on Windows) from the bin folder. The agent asks for an installation token while starting up. Provide the installation token you see in step 4 of this tutorial and press enter.

```
[Enter installation token :]
```

Once the agent successfully connects to Identity Cloud, a confirmation message is displayed on the command line.
You can further verify this by checking the Identity Cloud UI. It shows your agent is connected successfully to the Identity Cloud.

Your user-store is ready and now you can use the credentials of users in the connected LDAP to log in to the user portal and configure single sign-on for your configured application. You can connect multiple agents (only two at the moment) to cloud to achieve high availability for outbound agent.

**Configuring SSO for Custom Applications**

WSO2 Identity Cloud supports enabled SSO for applications that use standard authentication mechanisms such as SAML, OIDC etc. These applications are called ‘Custom Applications’. Before configuring SSO for any custom application, you have to first add a custom application. The following instructions describe how you can do that.

**Adding a Custom App**

1. Click on the menu bar on the top left corner.

2. The following left panel is displayed. Click on Applications.
3. Click **Add Application** on the following application landing page.

4. Click on **Custom Application** in the page that appears.

5. Enter **Application Name**, **Description**, and click **Add**.

Once you click **Add**, the following page is displayed with **Standard-based Federation** is selected by default.
as the App Type.

At this point, you can select the application type (Select App Type) of your custom application you wish to have SSO configured.

- Configuring Standards-based SSO
- Configuring Proxy-based SSO
- Configuring a Shortcut type Application

**Configuring Standards-based SSO**

WSO2 Identity Cloud allows SSO for custom applications that use various standard types such as SAML, OpenID Connect, and WS-Federation.

See the following topics for a description of the concepts that you need to know when configuring SSO for a standards-based custom application.

- How single sign-on works
- SAML 2.0 SSO
- OpenID Connect
- WS-Federation

The following diagram illustrates the process that is followed for authentication to an standards-based custom application once this configuration is done with the WSO2 Identity Cloud.
Figure: Accessing an standards-based custom application using Identity Cloud

The above diagram illustrates how a user accesses the standards-based custom application in the User Portal of the WSO2 Identity Cloud. The custom application sends an authentication request to the Identity Cloud using the protocol based on standard type (SAML, OpenID Connect or WS-Federation). The Identity Cloud sends an authentication response and the user is able to log in to the custom application.

In WSO2 Identity Cloud, configuring standards-based SSO can be done using the following ways:

- Configuring SAML SSO
- Configuring OpenID Connect SSO
- Configuring WS-Federation SSO

Configuring SAML SSO

Using WSO2 Identity Cloud, you can create a custom application and configure it to facilitate Single Sign-On (SSO) using SAML. This custom application is a representation of your web application and can be integrated with WSO2 Identity Cloud to provide SSO for users of this application.

**In this tutorial you will learn** how to configure single sign-on for your custom application using SAML.

To demonstrate SSO of a SAML-based custom application, first, you need to obtain and configure a web application. Next, you must configure a custom application with standards-type SAML to represent this web application. Users of the web application can access it using Single Sign-on via the Identity Cloud. This tutorial uses Travelocity as the sample web application to demonstrate how SSO works.

The following is a video of this tutorial that is done using Google. The steps in this tutorial differ from the video as this tutorial uses a sample web application.

See the following topics for instructions on how to configure the sample web application with WSO2 Identity Cloud.

- Adding a custom application
- Configuring the sample web application
- Configuring the custom application with the web app

Adding a custom application

To know how to add a custom application, see here.

Configuring the sample web application

To deploy and configure the travelocity web application, follow the steps below.

1. Download the travelocity.com.war file. Before you deploy the application, you may need to download
metadata.xml and edit the travelocity.properties file.

Expand to see how to download IdP metadata configuration xml

a. Log in to WSO2 Identity Cloud.
b. Click the menu icon on the top, left corner of the screen and click Applications:

![Identity Cloud / Applications](image)

Alternatively, click Overview on the menu bar and click View Applications.

c. Click **DOWNLOAD IDP METADATA** to download the IDP metadata file. (This file gets downloaded to a local folder.)

Expand to see how to edit the travelocity.properties file

a. Open command prompt.
b. Go to the folder where you have downloaded the travelocity.com.war file.
c. Create a new directory (e.g., `mkdir MyTest`).
d. Extract the `travelocity.com.war` to the newly created directory (e.g., `unzip travelocity.com.war -d MyTest/`). This extracts all the files to the `MyTest` directory.
e. Navigate to the `../WEB-INF/classes` directory.
f. Edit the `travelocity.properties` file (see sample below) found inside the `../WEB-INF/classes` directory.

Expand to see the sample of travelocity.properties.xml file

```
Sample of travelocity.properties.xml file
```

```
# Copyright (c) WSO2 Inc. (http://www.wso2.org) All Rights Reserved.
#
# WSO2 Inc. licenses this file to you under the Apache License,
# Version 2.0 (the "License"); you may not use this file except
# in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
```
EnableSAML2SSOLogin=true
EnableOpenIDLogin=true
EnableOAuth2SAML2Grant=false

#Url to do send SAML2 SSO AuthnRequest
SAML2SSOURL=samlsso

#Url to do initiate OAuth2 SAML2 Grant Request
OAuth2SAML2GrantURL=token

#Url to initiate OpenID Authentication Request
OpenIdURL=openid

#URIs to skip SSOAgentFilter; comma separated values
SkipURIs=/travelocity.com/index.jsp

#A unique identifier for this SAML 2.0 Service Provider application
SAML2.SPEntityId=travelocity.com

#The URL of the SAML 2.0 Assertion Consumer

#A unique identifier for this SAML 2.0 Service Provider application
SAML2.IdPEntityId=localhost

#The URL of the SAML 2.0 Identity Provider
SAML2.IdPURL=https://localhost:9443/samlsso

#Identifier given for the Service Provider for SAML 2.0 attributes
#exchange
SAML2.AttributeConsumingServiceIndex=1701087467

#Specify if SingleLogout is enabled/disabled
SAML2.EnableSLO=true

#This is the URL that is used for SLO
SAML2.SLOURL=logout

#Specify if SAMLResponse element is signed
SAML2.EnableResponseSigning=true

#Specify if SAMLAssertion element is signed
SAML2.EnableAssertionSigning=true
#Specify if SAMLAssertion element is encrypted
SAML2.EnableAssertionEncryption=false

#Specify if AuthnRequests and LogoutRequests should be signed
SAML2.EnableRequestSigning=true

#Password of the KeyStore for SAML and OpenID
KeyStorePassword=wso2carbon

#Alias of the IdP's public certificate
IdPPublicCertAlias=wso2carbon

#Alias of the SP's private key
PrivateKeyAlias=wso2carbon

#Private key password to retrieve the private key used to sign
#AuthnRequest and LogoutRequest messages
PrivateKeyPassword=wso2carbon

#OAuth2 token endpoint URL

#OAuth2 Client ID
OAuth2.ClientId=Qn5DQHCYfshxeZh6R9SL1HM21sMa

#OAuth2 Client Secret
OAuth2.ClientSecret=cbkAs1gajdwPAMbrSR54hPAIcz0a

#OpenId Provider Url
OpenId.ProviderURL=https://localhost:9443/openid/

#openid.return_to parameter
OpenId.ReturnToURL=http://localhost:8080/travelocity.com/home.jsp

#Custom SAML post binding request page
#SAML.PostBinding.RequestPage=path/to/the/html

#Additional request parameters
#QueryParams=tenantDomain=-1234

#SAML2.IsForceAuthn=true

#Specify whether the consumer requests user attributes from the provider
OpenId.EnableAttributeExchange=true
1. #Specify whether the consumer runs in dumb mode
   OpenId.EnableDumbMode=false

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML2.SPEntityId=travelocity.com</td>
<td>A unique identifier for this SAML 2.0 Service Provider application.</td>
</tr>
<tr>
<td>SAML2.AssertionConsumerURL=</td>
<td>The URL of the SAML 2.0 Assertion Consumer. Note: organization_domain should replace by actual tenant created.</td>
</tr>
<tr>
<td>SAML2.IdPURL=<a href="https://identity.cloud.wso2.com/identity/t/">https://identity.cloud.wso2.com/identity/t/</a>&lt;organization_domain&gt;</td>
<td>The URL of the SAML 2.0 Identity Provider. The value for SAML2.IdP URL is taken from 'SingleSignOnService' element in downloaded metadata xml file. (See sample below)</td>
</tr>
</tbody>
</table>

Sample of WSO2IdentityCloudMetadata.xml:

```
```

- g. Save the file.
- h. Navigate to the MyTest directory.
- i. Create the travelocity.com.war file (e.g., `jar -cvf travelocity.com.war *`).

2. Go to WSO2 Integration Cloud and deploy the travelocity web application. Click the following for instructions on how to deploy a web application in Integration Cloud.

   - Expand to see how a web application is deployed in Integration Cloud
   - a. Log in to Identity Cloud admin portal.
   - b. Click the Settings menu at the top right corner and navigate to Integration Cloud.
c. Click **Create Application**.

d. Scroll down and click on **Java Web Application** in **Services**.
e. Under **Deploy an artifact**, select the **Local File System** option and click **Continue**.

f. Enter an **Application Name** and click **Browse** to select the `travelocity.com.war` file.
You see the travelocity.com.war getting loaded.

![Upload a .WAR file](image1)

Once uploading is completed, click **Create** to deploy the application.

![Upload a .WAR file](image2)

![Create button](image3)

You see the deployment progress as shown below.

![Creating the application...](image4)

Once the application is successfully deployed, it is displayed as shown below.
Alternatively, you can see the deployed travelocity application in the Integration Cloud landing page.

Configuring the custom application with the web app

1. Once you enter the application name, the following page is displayed with Standards-based Federation and SAML2 Web SSO is selected by default as App type and Security Protocol. Make sure the Configure Manually button is selected.
2. Scroll down and enter **Issuer** as 'travelocity.com' and provide the **Assertion Consumer URL** as "http://demotenant-travelocity.wso2apps.com/travelocity.com/home.jsp" and click **Add** to add the link. Select **Enable Response Signing**.

3. Click **Save** to add the application details.
4. The added application is displayed, and you can delete or edit it by clicking the button next to the application thumbnail name.

Note that the tenant name *demotenant* should be replaced by the name of the tenant you have created.

5. Click **Go to User Portal**.
6. Click the added custom application that is in the user portal.

8. You are directed to travelocity home page (http://demotenant-travelocity.wso2apps.com/travelocity.com/home.jsp) without having to sign in explicitly. This shows SAML based SSO capability for a custom application using Identity Cloud.

Configuring OpenID Connect SSO

Using Identity Cloud, you can create a custom application and configure it to facilitate Single-Sign On (SSO) using OpenID Connect. This custom application can be integrated with the WSO2 Identity Cloud as indicated in this topic to provide SSO for users of this application.

In this tutorial you will learn how to configure single sign-on for your custom application using OpenID Connect.

Before you start this tutorial, you need to create a custom application in Identity Cloud. Click here to know how to add a custom application.

Once you add the application, the following page is displayed with Standards-based Federation and SAML2 Web SSO is selected by default as App Type and Security Protocol.

1. Select OpenID Connect from Select Security Protocol list.
2. Enter the **Redirect URL** as 'https://identity.clooudmgt:8080/playground2/oauth2client' and click **Generate Keys**.
The following page is displayed with the generated **OAuth Keys** and **OAuth Client Secret**.
3. Enter **Access URL**, the location that you want your custom application should be directed to.

![Store Configuration](image1)

4. Click **Save** to save the application details and the added application is displayed on the page.

![MyOAuthSampleApp](image2)

5. Click **Go to User Portal**.

![Go to User Portal](image3)

6. Click the added custom application that is in the user portal.
7. You are directed to WSO2 website home page without having to sign in explicitly. This shows OpenID Connect based SSO capability for a custom application using Identity Cloud.

Configuring WS-Federation SSO

Using Identity Cloud, you can create a custom application and configure it to facilitate Single-Sign On (SSO) using WS-Federation. This custom application can be integrated with the WSO2 Identity Cloud as indicated in this topic to provide SSO for users of this application.

In this tutorial you will learn how to configure single sign-on for your custom application using WS-Federation.

Before you start this tutorial, you need to create a custom application in Identity Cloud. Click here to know how to add a custom application.

Once you have created the custom application, the following page is displayed with Standards-based Federation and SAML2 Web SSO is selected by default. Now you need to configure the application as an Standard-based Federation type WS-Federation.

2. Enter the **Realm** as 'wso2' and **Wreply URL** as 'https://identity.cloud.wso2.com/myapp/service'.
3. In **Store Configurations**, enter the **Access URL** as ‘https://identity.cloud.wso2.com/myapp’.

4. Click **Save** to save the application details and the added application is displayed on the page.
5. Click **Go to User Portal**.

You are directed to WSO2 website home page without having to sign in explicitly. This shows WS-Federation based SSO capability for a custom application using Identity Cloud.

**Configuring Proxy-based SSO**
WSO2 Identity Cloud allows SSO for proxy-based applications using SAML 2.0. Proxy-type applications do not need to handle the SSO logic. There is a central gateway provided that acts as a proxy for the application, handles SSO requests/responses on behalf of the application, and sends a signed JWT token to the application where it can process it and identify the authenticated user.

**In this tutorial you will learn** how to configure single sign-on for your proxy-based SAML application.

See the following topics for a description of the **concepts** that you need to know when configuring SSO for a proxy-based SAML application.

- How single sign-on works
- SAML 2.0 SSO

The following diagram illustrates the process that is followed for authentication to a proxy-based SAML application once this configuration is done with the WSO2 Identity Cloud.

**Figure: Accessing a proxy-based SAML application using Identity Cloud**

The above diagram illustrates how a user accesses the proxy-based application from the application list in the *User Portal* of the WSO2 Identity Cloud. This is redirected to the Identity Cloud Gateway and an authentication request is sent to the Identity Cloud using the SAML protocol. The Identity Cloud sends an authentication response to the application through Identity Cloud Gateway and the user is able to log in to the application.

Now let's begin.

1. Select **Proxy-based Federation** from Select App Type.
2. Scroll down. In Gateway Configuration, enter Context as 'proxyContent' and Access URL as 'http://www.wso2.com'.

3. Click Save to save the application details and the added application is displayed on the page.
4. Click **Go to User Portal**.

5. Click the added custom application that is in the user portal.

You are directed to WSO2 website home page without having to sign in explicitly. This shows SAML based SSO capability for a proxy-based custom application using Identity Cloud.

**Configuring a Shortcut type Application**

WSO2 Identity Cloud allows creating shortcut type applications. These applications serve as shortcuts for some
websites and applications having own authentications etc., but should be listed in a central place with all other apps.

In this tutorial you will learn how to configure shortcut type application.

Now let's begin.

1. Select **Shortcut** from Select App Type.

2. In Store Configuration, enter **Display Name** and **Access URL**.
2. Click **Save** to add the shortcut app. The added app is displayed as shown below.

3. **Click Save** to add the shortcut app. The added app is displayed as shown below.

4. **Click Go to User Portal**.
4. Click the added shortcut application that is in the user portal.

5. You are directed to home page of the website. (i.e. https://www.nytimes.com/ as given in Access URL in Step 2)

Configuring SSO for Third-party Applications

WSO2 Identity Cloud allows configuring SSO for several third-party applications. These are listed in the following tutorials.

- Configuring SSO for Salesforce
- Configuring SSO for NetSuite
- Configuring SSO for Zuora
- Configuring SSO for AWS
- Configuring SSO for GoToMeeting
- Configuring SSO for Concur

Configuring SSO for Salesforce

Salesforce can be configured for SSO using WSO2 Identity Cloud by adding it as an application. After the configuration is done, you can simply access the Salesforce application from the applications list in the User Portal of WSO2 Identity Cloud. This triggers an authentication request to the Identity Cloud using the SAML protocol. Identity Cloud sends an authentication response, and you are able to log in to Salesforce. The following diagram illustrates this process:
Figure: Accessing a Salesforce application using WSO2 Identity Cloud

In this tutorial, you learn how to integrate Salesforce with WSO2 Identity Cloud to allow the users of Salesforce to automatically log in to the Identity Cloud.

See the following topics for a description of the concepts that you need to know when configuring SSO.

- How single sign-on works
- SAML 2.0 SSO

The following is a video of how this is done.

Before you begin, download the metadata XML file that allows you to set the identity cloud configuration details in any third-party application without having to key them in.

How to download IdP metadata configuration xml

1. Log in to WSO2 Identity Cloud.
2. Click the menu icon on the top, left corner of the screen and click Applications:

   ![Identity Cloud / Applications](image)

   Alternatively, click Overview on the menu bar and click View Applications.

3. Click DOWNLOAD IDP METADATA to download the IDP metadata file. (This file gets downloaded to a local folder.)

   ![Identity Cloud / Applications](image)
Let's get started!

- Setting up Salesforce for SSO
- Configuring WSO2 Identity Cloud for SSO with Salesforce

*Setting up Salesforce for SSO*

To setup Salesforce for SSO, do a domain deployment in Salesforce as follows:

   - How to create a developer account in Salesforce
     
     To create a developer account in Salesforce:
     
     
     2. Enter all your details and click ‘Sign me up’. Your developer account is created.
Get your very own Developer Edition

A full-featured copy of Force.com, for FREE.

Name
First
Last

Email
Your email address

Role
Developer

Company
Company Name

Country
Sri Lanka

Postal Code

Username
Ex: name@yourcompany.com

I have read and agree to the Master Subscription Agreement and the Privacy Statement.

Sign me up >
2. Register your mobile phone for verification purposes.

3. Enter the verification code you received on your mobile.
4. In the left navigation panel, in **SETTINGS/Company Settings** and click **My Domain**.

5. This is the Step 1 of the Domain Setup. Give a unique domain name and click **Check Availability** to make sure that the given domain name is unique.
The following is displayed if the domain is available for use.

6. Register the domain by clicking **Register Domain**. Once the domain is registered, you get an email indicating that the domain is registered and ready for testing.
Once you click the **Register Domain**, you are taken to the second step of the domain setup process. You will receive an email from Salesforce.

7. Click the link in the email that directs you to the Salesforce login page.
8. Log in to Salesforce using your username and password that were created in step 1.
9. In the left navigation panel, in **Company Settings** under **SETTINGS**, click **My Domain**.
You are directed to the third step of the domain setup process.

10. Click the **Deploy to Users** button to make the domain available for users.

11. Click **OK** to confirm the domain deployment.
Once you click OK, you are directed to step 4 of the domain setup process. This completes the deployment process. The next step is to configure SAML settings for single sign-on.

12. In the left navigation panel, in SETTINGS, expand Identity and select Single Sign-On Settings.
13. In the **Single Sign-On Settings** page, click **Edit**.

14. In **Federated Single Sign-On Using SAML**, click the **SAML Enabled** check box and click **Save**.
15. Click **New from Metadata File** to upload the metadata file that you downloaded in the **Before you begin** section above.

16. Click **Choose File** to select the metadata file.

17. Once you select the metadata file, click **Create** to load the Identity Cloud's metadata details to Salesforce.

18. Verify the details and click **Save**.
The following page appears with the SAML settings that you configured for SSO with WSO2 Identity Cloud.

Entity ID and Endpoints fields are used when configuring Salesforce in WSO2 Identity Cloud. (see step 7 in the Configure WSO2 Identity Cloud for SSO with Salesforce section.)

**Configuring WSO2 Identity Cloud for SSO with Salesforce**

1. Log into WSO2 Identity Cloud.
2. Click the menu icon on the top, left corner of the screen.
3. Click **Applications** from the *Admin Portal* to navigate to the Application list.

4. Click **ADD APPLICATION** to add a Salesforce application.

5. Select **Salesforce** icon.

6. Enter an **Application Name** and click **Add**.
7. Enter **Issuer** and **Assertion Consumer URL** and click **Add**. For these fields use **Entity ID** and **Salesforce login URL** respectively. (See **Configure SSO in Salesforce**, step 18)

8. In Store Configuration, enter **Display Name**, **Access URL** (same as **Issuer**), and click **Save**.
The added Salesforce app is displayed in **Identity Cloud/Applications page**.

9. Once the application is added, it is listed in **User Portal**. Click **Go to User Portal** at the top right corner of the page.
10. Click the added Salesforce app.

Now you can access the Salesforce home page without having to sign in because you configured SSO between Salesforce and WSO2 Identity Cloud.

Configuring SSO for NetSuite

NetSuite can be configured for SSO using WSO2 Identity Cloud by adding it as an application. After the configuration is done, you can simply access the NetSuite application from the applications list in the User Portal of WSO2 Identity Cloud. This triggers an authentication request to the Identity Cloud using the SAML protocol. Identity Cloud sends an authentication response, and you are able to log in to NetSuite. The following diagram illustrates this process:
In this tutorial, you learn how to integrate NetSuite with WSO2 Identity Cloud to allow the users of NetSuite to automatically log in to the Identity Cloud.

See the following topics for a description of the concepts that you need to know when configuring SSO.

- How single sign-on works
- SAML 2.0 SSO

Let's get started.

- Setting up NetSuite for SSO
- Configuring WSO2 Identity Cloud for SSO with NetSuite

**Setting up NetSuite for SSO**

2. In the Home page, click Setup and select Setup Manager.

3. In the Setup Manager left navigation menu, expand Integration and select SAML Single Sign-on listed under Integration Management.
4. In the SAML Setup page, provide the following values:
   - For LOGOUT LANDING PAGE, enter 'https://identity.cloud.wso2.com/user-portal/t/<tenant-name>'. For example, this can be https://identity.cloud.wso2.com/user-portal/t/tenant10.com.

5. Upload the metadata from the local folder and click Submit.
6. Configure an on-premise user store for NetSuite. **NetSuite needs a special claim that denotes the company ID.** So each user should have the ID as a user attribute. When you download the agent, a claim-config.xml is created. This is created in the `<ON_PREMISE_AGENT_HOME>/conf/` directory. Add the following entry to the config file.

```xml
<Claim>
  <ClaimURI>http://wso2.org/claims/netsuiteid</ClaimURI>
  <AttributeID>{AN_LDAP_ATTRIBUTE}</AttributeID>
</Claim>
```

The value of the attribute should be your company ID in NetSuite. e.g. 3883026

**Configuring WSO2 Identity Cloud for SSO with NetSuite**

1. Log in to WSO2 Identity Cloud.
2. Click the menu icon on the top, left corner of the screen.
3. Click **Applications** from the **Admin Portal** to navigate to the **Application list**.
4. Click **ADD APPLICATION** to add a NetSuite application.

5. Click **NetSuite** icon.

6. Enter an **Application Name** and click Add.
6. Click Add Application and in SSO Configuration, select 'https://system.na1.netsuite.com/saml2/acs' as the default Assertion Consumer URL.

7. Click Add Application and in SSO Configuration, select 'https://system.na1.netsuite.com/saml2/acs' as the default Assertion Consumer URL.

8. In Store Configuration, enter a Display Name, and click Save.
The added NetSuite app is displayed in Identity Cloud/Applications page.

9. Once the application is added, it is listed in User Portal. Click Go to User Portal at the top right corner of the page.
9. Click the added NetSuite App.

Now you can access the NetSuite home page without having to sign in because you configured SSO between NetSuite and WSO2 Identity Cloud.

**Configuring SSO for Zuora**

**Configuring SSO for AWS**

Amazon web services (AWS) can be configured for SSO using WSO2 Identity Cloud by adding it as an application. After the configuration is done, you can simply access the AWS application from the applications list in the User Portal of WSO2 Identity Cloud. This triggers an authentication request to the Identity Cloud using the SAML protocol. The Identity Cloud sends an authentication response, and you are able to log in to AWS. The following diagram illustrates this process:
In this tutorial you will learn how to integrate AWS with WSO2 Identity Cloud to allow the users of AWS to automatically log in to the Identity Cloud.

See the following topics for a description of the concepts that you need to know when configuring SSO.

- How single sign-on works
- SAML 2.0 SSO

Before you begin, download the metadata XML file that allows you to set the identity cloud configuration details in any third-party application without having to key them in.

How to download IdP metadata configuration xml

1. Log in to WSO2 Identity Cloud.
2. Click the menu icon on the top, left corner of the screen and click Applications:

Alternatively, click Overview on the menu bar and click View Applications.

3. Click DOWNLOAD IDP METADATA to download the IDP metadata file. (This file gets downloaded to a local folder.)

Let's get started!

- Setting up AWS for SSO
- Configuring WSO2 Identity Cloud for SSO with AWS

Setting up AWS for SSO

1. Go to https://aws.amazon.com and click Sign in to the Console.
2. Sign in to AWS Console using a valid AWS account.

3. In the AWS Services page, under Security, Identity & Compliance, click IAM.

4. In the left navigation panel, click Identity providers.
5. Click **Create Provider**.

6. Create an identity provider by selecting the provider type as **SAML**, entering a **Provider Name**, uploading IDP metadata xml file, and clicking **Next Step**.
7. Verify the **Provider information** and click **Create**.

Once the Identity Provider is created, you see the following screen with the message that the SAML provider is created and the created provider is listed with **Type** of protocol and **Creation Time**.
8. Now, you need to configure a role for SSO. In the left navigation panel, click Roles.

9. Click Create new role.

10. In Select role type screen, select Role for identity provider access option and select Grant Web Single Sign-On (WebSSO) access to SAML providers by clicking Select button.
11. In **Establish Trust** page, select the SAML provider that you have creating the role for (i.e. wso2_identity_cloud) and click Next Step.


14. In Set role name and review page, provide a valid role name and click Create Role.
14. Once the role is created, you can see it is listed with its name, description and creation time.

15. The next step is to configure an on-premise user store for AWS. Since AWS needs a special claim to help them decide the permissions of the signing in user, the following changes should be done in the `<ON_PREMISE_AGENT_HOME>/conf/claim-config.xml` file. This file is created when you download the agent.

```
<Claim>
  <ClaimURI>http://wso2.org/claims/awsrole</ClaimURI>
  <AttributeID>{AN_LDAP_ATTRIBUTE}</AttributeID>
</Claim>
```

AWS LDAP Settings

It is required at the AWS end to have an LDAP attribute set for the users.
The value of the attribute should be `<AWS_SSO_ROLE_ARN>,<AWS_SSO_IDP_ARN>`
e.g.

Configuring WSO2 Identity Cloud for SSO with AWS

1. Log in to WSO2 Identity Cloud.
2. Click the menu icon on the top, left corner of the screen.

3. Click Applications from the Admin Portal to navigate to the Application list.

4. Click ADD APPLICATION to add a AWS application.

5. Select AWS icon.

6. Provide an application name and click Add.
7. In **Store Configuration**, provide a **Display name**, and click **Save**.

The added AWS app is displayed in **Identity Cloud/Applications page**.
8. Once the application is added, it is listed in User Portal. Click Go to User Portal at the top right corner of the page.

9. Click the added AWS App.

Now you can access the AWS home page without having to sign in because you configured SSO between AWS and WSO2 Identity Cloud.

**Configuring SSO for GoToMeeting**

GoToMeeting supports single sign-on (SSO) that is initiated by the identity provider. GoToMeeting can be integrated with the WSO2 Identity Cloud as indicated in this topic to provide SSO for users of this application.
In this tutorial you will learn how to configure single sign-on for your GoToMeeting application.

Click the following topics for a description of the concepts that you need to know when configuring SSO:

- How single sign-on works
- SAML2.0 SSO

1. Log in to the WSO2 Identity Cloud Admin Portal.
2. Click on the menu bar on the top left corner and click Applications.
3. Click Download IdP Metadata. The identity provider metadata file will be downloaded on to your machine.
4. Login to the GoToMeeting admin console and select the Identity provider tab.
5. Select Upload SAML metadata file and upload the metadata file you downloaded in step 3. For more information, see the GoToMeeting Support Documentation.

6. Click Save. You will be redirected to a screen similar to one below. Verify the details and click Save again.
You have now set up the identity provider in GoToMeeting. Next, configure the WSO2 Identity Cloud for SSO.

7. On the WSO2 Identity Cloud, click **Applications** on the menu bar and then click **Add Application**.

8. Select the **GoToMeeting** tile.
9. Enter an **Application Name** and **Description** for the application and click **Add**.

10. Enter the application details.
10. Click to view more information about application details

a. **Issuer**: This is the GoTo SAML Service’s entityID. Leave this as the default value ([https://login.citrixonline.com/saml/sp](https://login.citrixonline.com/saml/sp))

b. **Assertion Consumer URL**: This is the URL authentication responses (containing assertions) are returned to. Select the assertion consumer URL from the provided list depending on the GoTo products that require SSO access:
   - [https://login.citrixonline.com/saml/app.gotoassist.com/acs](https://login.citrixonline.com/saml/app.gotoassist.com/acs) (GoToAssist)
   - [https://login.citrixonline.com/saml/global.gototraining.com/acs](https://login.citrixonline.com/saml/global.gototraining.com/acs) (GoToTraining)
   - [https://login.citrixonline.com/saml/global.gotowebinar.com/acs](https://login.citrixonline.com/saml/global.gotowebinar.com/acs) (GoToWebinar)
   - [https://login.citrixonline.com/saml/acs](https://login.citrixonline.com/saml/acs) (Global - All products)
   - [https://login.citrixonline.com/saml/global.gotomeeting.com/acs](https://login.citrixonline.com/saml/global.gotomeeting.com/acs) (GoToMeeting)

c. **Display Name**: Enter a display name for your application.

d. **Tags**: Enter any keywords or tags for searching purposes.

e. **Visibility**: Enter role names to define which roles have SSO access to the application.

11. Click **Save**.

You have now successfully set up SSO for the GoToMeeting application.
Try it out

1. Login to the WSO2 Identity Cloud User Portal.
2. Click on the GoToMeeting application. You will be automatically logged in to your GoToMeeting account via the WSO2 Identity Cloud.

Configuring SSO for Concur

This page is a work in progress as these steps are being tested from the Concur side.

Concur only supports single sign-on (SSO) that is initiated by the identity provider. Concur can be integrated with the WSO2 Identity Cloud as indicated in this topic to provide SSO for users of this application.

In this tutorial you will learn how to configure single sign-on for your Concur application.

See the following topics for a description of the concepts that you need to know when configuring SSO for a Concur application.

- How single sign-on works
- SAML 2.0 SSO

The following diagram illustrates the process that is followed for authentication to a Concur application once this configuration is done with the WSO2 Identity Cloud.

Figure: Accessing a Concur application using Identity Cloud

The above diagram illustrates how a user accesses the application from the application list in the User Portal of the WSO2 Identity Cloud. This is redirected to the Concur application and an authentication request is sent to the Identity Cloud using the SAML protocol. The Identity Cloud sends an authentication response and the user is able to log in to Concur.

Before you begin you need to get in touch with a technical contact from Concur who can help to setup your certificate and SAML activation process from the Concur side. This is a manual step that cannot be done via online means. You MUST get the help of the technical contact from Concur for this to be done.

Once you are in touch with the technical contact from Concur, you can get started and set this up.

To integrate WSO2 Identity Cloud with Concur:

1. Log in to the Identity Cloud.
2. Click on the menu bar on the top left corner and click Applications.
3. Click the Add Application button to add an application.

4. Click Concur from the application list that appears and click Continue.

5. Fill in a suitable Application Name and Description and click Add.
6. In the form that appears, provide a **Display Name**, and click **Add** to add the application.
The Concur application is displayed in the application list.

7. Click **User Portal** at the top right corner of the page.
8. Click the added Concur application.

9. You are directed to Concur home page without having to sign in explicitly.

### Configuring Custom Themes

In WSO2 Identity Cloud, you can add your own custom themes to the User Portal. This involves easily changing the color of the theme and the company logo on the header. Identity Cloud provides you with a user interface to upload or delete themes from the Admin Portal.

**In this tutorial,** you configure custom themes for the User Portal using Identity Cloud Admin Portal.

The following is a video of how this is done.

---

**Before you begin**

Before configuring custom themes, create a customized theme as a .zip file by maintaining the following file structure and file names. Also make sure your company logo image is in .png format.

```
|--- css/
    |--- appm-left-column-styles.css
    |--- appm-main-styles.css

|--- img/
    |-- logo.png
```

For this tutorial, the custom-theme file `ABC-user-portal-theme.zip` is used to demonstrate the feature. You can also use `default-user-portal-theme.zip` for any further customizing.
Now, let’s begin.

1. Log in to the Admin Portal and click User Portal Themes from left menu.

2. Provide a Theme Name, click Browse to upload the .zip file (the .zip file that you created including CSS files and your logo), and click Save. The theme is applied to the User Portal.

3. Navigate to the User Portal by either clicking the user portal link in the message or the Go to User Portal button on upper right corner of the page.
You can see your new theme (i.e: ABC as company name) is applied in the User Portal.

If you need to change the theme, delete the current theme and upload a new theme. You can also download the current theme from the Admin Portal and do any changes and upload it back.

**Note:** If you are doing any changes to the contents of the downloaded ABC-user-portal-theme.zip file (i.e., replacing the logo), make sure you compress the contents to make the .zip file before uploading.

### Configuring Custom User Portal URL

In WSO2 Identity Cloud, the default URL for your User Portal takes the format `https://identity.cloud.wso2.com/user-portal/t/<tenant>`. For example, if your tenant name is ‘foo’, the default URL of your User Portal is `https://identity.cloud.wso2.com/user-portal/t/foo`. 
You can use a customized URL that is more representative of your company or personal branding instead of using the default URL.

In this tutorial, you learn how to generate SSL certificates and DNS records to configure a custom URL for WSO2 Identity Cloud.

Let's begin.

- Create SSL certificates and DNS records
- Customizing the User Portal URL

Create SSL certificates and DNS records

1. Install a SSL key generation tool (OpenSSL is used in this tutorial).
2. Using the command-line, navigate to a location of your choice in the server and execute the following command to generate a private SSL key by the name private.key.

   ```bash
   openssl genrsa -out private.key 2048
   ```

   Note that the key file is generated in your folder location as 'private.key'.

3. In the command-line, execute the following command to generate a certificate-signing-request file for your custom URL. Make sure you change the business address in this command to your own.

   ```bash
   openssl req -new -key private.key -sha256 -nodes -out request.csr -subj "/C=US/ST=California/L=Mountain View/O=WSO2/OU=IT/CN=developers.mytesturl.info"
   ```

   Note that the certificate-signing-request file is generated in your folder location as 'request.csr'.

4. Go to a certificate vendor of your choice and use the certificate-signing-request file to obtain a certificate for your domain.

Any certificate that is accepted by the browser should work. In this tutorial, https://www.comodo.com/ is used as the certificate authority.

When you are done, you receive an email with the certificate for your domain along with the certificate authority's root and intermediate certificates.

Some certificate authorities provide the root and intermediate files as a single chain file, while others provide multiple files.

If you receive multiple root and intermediate files from your certificate authority, use the cat utility (available in Unix and Unix-based operating systems) to concatenate them to a single chain file (chain.crt). For example:

   ```bash
   cat COMODORSADomainValidationSecureServerCA.crt COMODORSAAAddTrustCA.crt AddTrustExternalCARoot.crt > chain.crt
   ```

Tip: If you are using Microsoft Windows, do the following to concatenate the certificate files:
Note that the chain.crt file should have content in the following order:

```
-----BEGIN CERTIFICATE-----
(Your Intermediate certificate: COMODORSADomainValidationSecureServerCA.crt)
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
(Your Intermediate certificate: COMODORSAAAddTrustCA.crt)
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
(Your Root certificate: AddTrustExternalCARoot.crt)
-----END CERTIFICATE-----
```

5. Reserve a domain name with any domain registrar and create DNS CNAME records that map the domain to your Identity user portal.

**Tip:** Most domain registrars provide step-by-step instructions in their websites. For your convenience, the general steps are listed below:

- Sign in to the domain registrar’s site.
- Navigate to your Domain Name Server (DNS) management page. The location and name of this page vary by the host but can generally be found under the 'Domain Management' or 'Advanced Settings' section.
- Find the CNAME settings. Under the 'CNAME value or alias', enter the subdomain that you would like to map each URL to. The subdomain of developers.mytesturl.info is developers’. 
- Set the CNAME destination to the Identity Cloud's custom DNS endpoint, which is customdns.identity.cloud.wso2.com.

Now, you have the SSL certificates and DNS records that you need to configure a custom URL for the Identity Cloud.

**Customizing the User Portal URL**

Subscribers enter the User Portal URL into a browser to get to your User Portal. Follow the steps below to customize the URL of your User Portal:

1. Log in to the Identity Cloud as the tenant admin.
2. Click the Setting icon at the top, right hand corner and select Custom URL (under Manage your cloud) from the options that appear.
3. Select the **Identity Cloud** tab and click **Modify** to change the existing domain.

4. Click **Verify** to check whether a CNAME record exists for this URL.
If the CNAME verification is successful, a screen is prompted for the SSL certificates.

5. Upload the files (SSL Certificate, SSL Key File and Chain File) that you created and click Proceed.
   - Click here to see the certificate files requirements
   - The certificate files must satisfy the following requirements:
### File Requirements

**SSL certificate**
This is the certificate that you got in step 4 of section ‘Create SSL certificates and DNS records’. It must satisfy the following requirements:

- In X509 format
- Not self signed
- Not expired
- Issued directly or by a wild card entry for a provided custom URL. For example:
  - In the direct method, if the CNAME is identitycloudtest.wso2.com, the issued SSL file must contain identitycloudtest.wso2.com.
  - In the wildcard method, if the CNAME is identitycloudtest.wso2.com, the issued SSL file should be *.wso2.com.

**SSL Key File**
This is the private key of the certificate that you got in step 2 of section ‘Create SSL certificates and DNS records’. It must be encrypted in the RSA format.

**Chain File**
This is the public key of the certificate that you got in step 4 of the section ‘Create SSL certificates and DNS records’. If the public key is included in the SSL file, extract it to a chain file.

---

If the files are successfully uploaded, you receive a notification saying “Custom URL mapping is successfully added”.

---

![User Portal screenshot](image-url)
Tip: Wait approximately 10 minutes for the changes to take effect. Adding the configurations and restarting the load balancers can take some time.

You have now successfully changed the user portal domain name to a custom value.

Try it out

Access the User Portal using your new URL. In this example, the new user portal URL is https://www.mytesturl-identity.wso2stagingapps.com.
Cloud Administration

Click a topic to see how to perform the task:

[ Add a new organization ]  [ Customize your custom portal URLs ]  [ Add members to your organization ]  [ Register and invite members ]  [ Customize Invitation Emails ]

Add a new organization

If you have already signed up to WSO2 Cloud and you want to add a new organization using the same email address, do the following:

1. Log in to the organization that you are already registered to in WSO2 Cloud.
2. Click the settings icon in the upper, right-hand corner of the UI and click **Organization**.
3. In the **Manage Organizations** page that opens, click **Add Organization** to create a new organization.

Customize your custom portal URLs

To change the custom portal URLs to something that is more representative of your company or personal branding, click the settings icon in the upper, right-hand corner of the UI and click **Custom URL**.
For a detailed tutorial, see Configuring Custom User Portal URL.

Add members to your organization

When you register to the Cloud, you become the admin of your organization. You can add new members and assign them different roles. Click the settings icon in the upper, right-hand corner of the UI and click Members.
Register and invite members

WSO2 offers different public, private and hybrid cloud solutions and all of them are collectively known as WSO2 Cloud (http://cloud.wso2.com). You can register to all of the public cloud offerings (e.g., the Identity Cloud, App Cloud) using a valid e-mail address. For a description of the Cloud’s user model, see Cloud’s user model.

In this tutorial, you register your organization in the Identity Cloud and invite members to it.

Members are added to the Cloud in 2 steps:

Let’s get started.
1. Go to http://cloud.wso2.com and sign up to any one of the public clouds. The UI guides you through the signup process.
2. Log in to the organization that you created. You have admin rights to it.
3. Click the settings icon in the top right-hand corner of the screen and then click the **Members** menu.

![Navigate to Cloud](image)

**Tip**: You can create multiple different organizations under the same login using the **Organization** menu, which is next to the **Members** menu.

4. Click **Invite Members**.

![Members](image)

5. Give valid e-mail addresses of the members, specify the role and invite. (Identity Cloud has only admin role listed under Common category)
5. Note that the member is sent an invitation e-mail, which has to be accepted by them in order to be added to the Cloud as a registered member.

You registered an organization and invited its members into the Cloud.

Customize invitation emails

WSO2 Cloud enables you to customize the emails sent to users.

You can customize the emails sent to the invited users by adding your own logo and changing the content in the emails.

If you want to make your own customizations as demonstrated below, contact WSO2 Identity Cloud Support via a support request or chat.

Admin Role

The Identity Cloud has only the Admin role. Any user with Admin role privileges, can create new users and roles, and configure permissions for the roles.

6. Note that the member is sent an invitation e-mail, which has to be accepted by them in order to be added to the Cloud as a registered member.

You registered an organization and invited its members into the Cloud.

Customize Invitation Emails

WSO2 Cloud enables you to customize the emails sent to users.

You can customize the emails sent to the invited users by adding your own logo and changing the content in the emails.

If you want to make your own customizations as demonstrated below, contact WSO2 Identity Cloud Support via a support request or chat.
The default invitation email, with the WSO2 logo and graphics, sent to the users with Admin role is in the following format:
WSO2 Identity Cloud provides you a one-time link, which will be sent with the invitation. You can customize the content of the email and choose where the one-time link should be included. The fields for the roles will be replaced according to your selection. You can also include your own logo and graphics to customize the email further.

An example of a customized email is shown below:

Hi,

You have been invited to join MyOwnCompany1, which is an organization powered by WSO2 Cloud. You have been assigned the Admin role.

Click this one-time link to log in: https://cloudmgt.cloud.wso2.com/cloudmgt/site/pages/confirm-verification.jag?confirmation=3dd8a13e-c12e-43a0-cfcf-1e5f02ea2f73&isInvitee=true

If you have any questions or need help, please contact us at cloud@wso2.com or simply reply to this message.

Regards,
WSO2 Cloud Team
Hi,

You have been invited to the ABC organization. You have been assigned the Admin role.

Click this one-time link to log in:
https://cloudmgt.cloud.wso2.com/cloudmgt/site/pages/confirm-verification.jag?confirmation=04fde95c-a836-4bd7-d66b-527f52bd39b4&isInvitee=true

If you have any questions or need help, please contact us at admin@abc.com

Getting Started Guide:
https://docs.wso2.com/display/APICloud/WSO2+API+Cloud+Documentation

Regards,
ABC

**Note:** These invitation emails will have the subject "WSO2 Cloud Services". If you need to customize the email subject, please mention it in your request as well.
FAQ

The following are some frequently asked questions pertaining to WSO2 Identity Cloud.

- **General FAQs**
  - How can I register to use Identity Cloud?
  - What happens if Identity Cloud goes down?
- **Identity Cloud-specific FAQs**
  - What is Identity Cloud?
  - Why should I use Identity Cloud?
  - How can I configure a new App with Identity Cloud?
  - What are the supported security protocols?
  - Do I need to share LDAP credentials with Identity Cloud, if I need to give application access to organization users?
  - Is it secure to use on-premise-user-store agent?
  - Do I need to remember or bookmark application URLs that I frequently access?

**General FAQs**

*How can I register to use Identity Cloud?*

*What happens if Identity Cloud goes down?*

**Identity Cloud-specific FAQs**

*What is Identity Cloud?*

*Why should I use Identity Cloud?*

*How can I configure a new App with Identity Cloud?*

*What are the supported security protocols?*

*Do I need to share LDAP credentials with Identity Cloud, if I need to give application access to organization users?*

*Is it secure to use on-premise-user-store agent?*

*Do I need to remember or bookmark application URLs that I frequently access?*