

# Single Sign-on SSO with SAML

This guide will show you how to set up SSO with SAML 2.0 and Kiuwan.

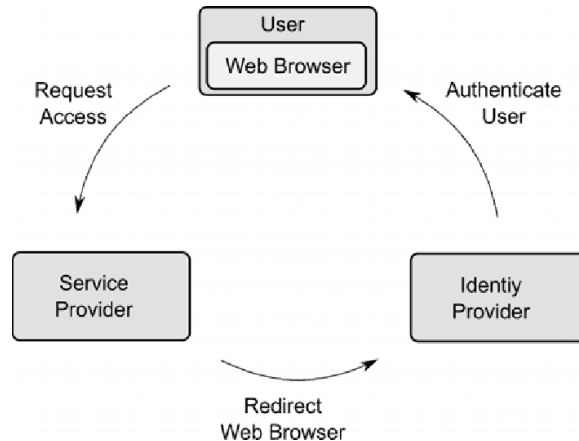
## Contents:

- [Introduction](#)
- [What is SAML?](#)
  - [SAML Security requirements](#)
- [Web Browser Single Sign-On](#)
  - [SAML 2.0 Metadata](#)
- [How to configure Kiuwan to work with SSO - SAML](#)
  - [Kiuwan configuration: How to configure your IdP in Kiuwan](#)
  - [IdP configuration: How to configure Kiuwan as Service Provider](#)
    - [Active Directory Federation Services \(ADFS\) configuration](#)
- [How to log into Kiuwan in a Web SSO scenario](#)
- [How to configure Kiuwan clients to work with SSO - SAML](#)
  - [Kiuwan Local Analyzer \(KLA\): SSO configuration](#)
  - [Kiuwan for Developers \(K4D\): SSO configuration](#)
  - [REST-API: SSO configuration](#)
- [SSO login vs username-password login](#)
  - [Adding a new user in an SSO-enabled account](#)
- [Appendix - Azure Active Directory configuration: How to configure Kiuwan as Service Provider](#)
  - [Login from the Kiuwan site](#)

## Introduction

In a **SAML - SSO scenario**, we can define the following **actors** or participants:

- A **User** requesting for some resource or service
- A **Service Provider (SP)** that receives the request and provides the service or access to the resource
- An **Identity Provider (IdP)** that authenticate the user and asserts the user identity



SSO can be implemented through different protocols, with SAML and OpenId Connect being the most widely used.

**Kiuwan currently supports SAML.** This document serves as a how-to to use Kiuwan in an SSO-SAML environment.

In summary, if your organization is using some kind of centralized user credentials repository implementing SAML and you want to use those enterprise credentials to authenticate in Kiuwan, this document provides you with information on how to set up Kiuwan to participate in an SSO-SAML environment.

## What is SAML?



**SAML** stands for **Security Assertion Markup Language** and it's an **open standard for exchanging authentication and authorization data between parties**. In particular, **between an identity provider (IdP) and a Service Provider (SP)**.

SAML is an XML-based markup language for security **assertions** usually transferred from IdPs to SPs. These assertions are used by SPs to make access-control decisions.

SAML assertions **contain three types of statements**:

### 1. Authentication statements

- Example: User U has been successfully authenticated at time T using method M of authentication

### 2. Attribute statements

- Example: User U does contain value V for attribute A

### 3. Authorization statements

- Example: User U is permitted to perform action A on resource R

Besides assertions, SAML defines **SAML protocols**, i.e. the processing rules to use assertions between SPs and IdPs.

Examples of such protocols are :

- Assertion Query and Request Protocol
- Authentication Request Protocol
- etc.

These SAML protocols can be mapped to standard **messaging formats**. This mapping is called a **SAML binding**. Examples of such bindings include:

- SAML SOAP Binding
- HTTP Redirect (GET) Binding
- HTTP POST Binding
- etc.

Finally, **SAML profiles** describe in detail how SAML assertions, protocols, and bindings combine to support a defined use case.

SAML 2.0 provides support for many profiles such as:

- Web Browser SSO Profile
- Identity Provider Discovery Profile
- Assertion Query/Request Profile
- etc



**The most important SAML 2.0 profile is the Web Browser SSO Profile, and it's fully supported by Kiuwan.**

## SAML Security requirements

The SAML specifications recommends:

- TLS 1.0+ for transport-level security
- XML Signature and XML Encryption for message-level security

# Web Browser Single Sign-On

Here is an image describing how Single Sign-On works:

Image	Description
<pre>sequenceDiagram     participant SP as Service Provider     participant UA as User Agent     participant IDP as Identity Provider      SP-&gt;&gt;UA: 1 Request target resource     UA--&gt;&gt;SP: (Discover the IdP)     SP-&gt;&gt;UA: 2 Redirect to SSO Service     UA-&gt;&gt;IDP: 3 Request SSO Service     IDP--&gt;&gt;UA: (Identify the user)     IDP--&gt;&gt;UA: Respond with XHTML form     UA-&gt;&gt;SP: 4 Request Assertion Consumer Service     SP-&gt;&gt;UA: 5 Redirect to target resource     UA-&gt;&gt;SP: 6 Request target resource     SP-&gt;&gt;UA: 7 Respond with requested resource     UA-&gt;&gt;SP: 8</pre> <p>The diagram illustrates the Web Browser Single Sign-On process involving three entities: Service Provider (SP), User Agent (UA), and Identity Provider (IDP). The process is numbered 1 through 8:</p> <ol style="list-style-type: none"><li>1. The user (usually through a web browser) requests a resource to a Service Provider (SP).</li><li>2. If a valid security context does not exist, the SP redirects the user agent to the Identity Provider's (IdP) SSO Service.</li><li>3. The user agent issues a request to the IdP's SSO Service to identify the user (if there's not a previous security context).</li><li>4. IdP validates the request and responds to the user agent.</li><li>5. The user agent sends the "authentication" assertion to the SP.</li></ol>	<ol style="list-style-type: none"><li>1. The user (usually through a web browser) requests a resource to a Service Provider (SP)</li><li>2. If a valid security context does not exist, the SP redirects the user agent to the Identity Provider's (IdP) SSO Service</li><li>3. The user agent issues a request to the IdP's SSO Service to identify the user (if there's not a previous security context)</li><li>4. IdP validates the request and responds to the user agent</li><li>5. The user agent sends the "authentication" assertion to the SP</li></ol>

6. The SP processes the assertion and redirects the user agent to the requested resource
7. The user agent requests SP for the requested resource
8. Finally, the SP returns the resource to the user agent.

## SAML 2.0 Metadata

In the Web Browser SSO workflow above, there are some interactions between the IdP and the SP that are based on mutual trust, for example:

- How does the SP know the IdP is authentic? And in turn, how does the IdP know the SP is authentic?
- How does the SP know where to send the user agent with the auth request? And how does the IdP know where to send the user agent with the auth response?
- How does the IdP encrypt the SAML assertion so that the trusted SP (and only the trusted SP) can decrypt the assertion?
- How does the service provider know that the auth response is coming from a trusted IdP?



These and other similar *trust* conditions are based on the use of **SAML 2.0 Metadata**.

Metadata ensures a secure transaction between an IdP and an SP through the sharing of trusted information.

SAML 2.0 provides a well-defined, interoperable metadata format that entities can leverage to bootstrap the trust process.

Regarding SSO SAML actor's identity, **metadata are defined for**:

- **Identity Provider metadata** (to publish identifying information about the IdP itself)
- **Service Provider metadata** (to publish identifying information about the SP itself)

Also, the **endpoints of communication** are defined by metadata, such as:

- **SSO Service metadata** (description of IdP's SSO endpoint)
- **Assertion Consumer Service** (desc of SP's service to send assertions from the IdP)

## How to configure Kiuwan to work with SSO - SAML



As explained before, **Kiuwan plays the role of Service Provider (SP) in an SSO - SAML context**.

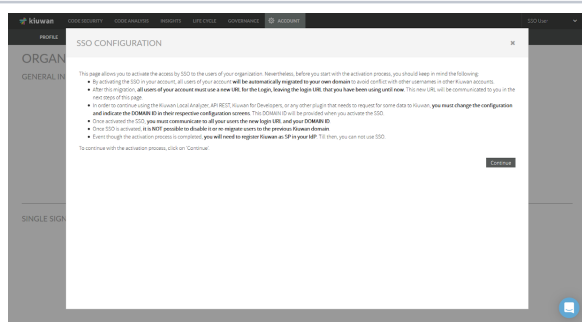
To configure SSO in Kiuwan you must first, of course, rely on an existing Identity Provider (IdP). There are many available IdP systems, all of them sharing SAML concepts (more or less adapted to their terminology).

As seen above, to set up a Web SSO environment, **SAML agents (IdP and SP) need to be identified and let each other know of their existence**.

This step is accomplished by **exchanging each other's metadata.**

## Kiuwan configuration: How to configure your IdP in Kiuwan

Go to **Account Management > Organization** and click **Configure SSO**.

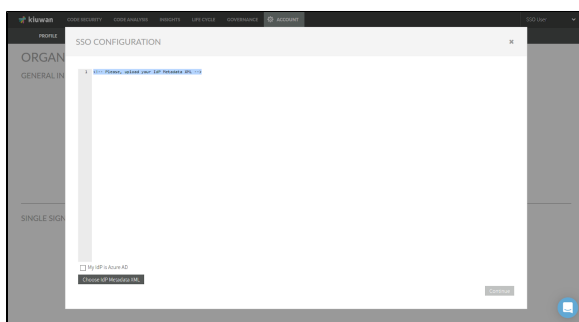
Image	Description
	<p>The following notes are shown in the window, which should be read carefully:</p> <ul style="list-style-type: none"><li>By activating the SSO in your account, all users of your account will be automatically migrated to your domain ID. To avoid conflict with other usernames in other Kiuwan accounts.</li><li>After this migration, all users of your account must use a new URL for the Login, leaving the login URL that you have been using until now. This new URL will be communicated to you in the next step of this page.</li></ul>

- To continue using the **Kiuwan Local Analyzer**, **API REST**, **Kiuwan for Developers**, or any other plugin that needs to request for some data to Kiuwan, **you must change the configuration and indicate the DOMAIN ID** in their respective configuration screens. This DOMAIN ID will be provided when you activate the SSO. (see further sections on these topics)
- Once activated the SSO, you must **communicate to all your users the new login URL and your DOMAIN ID**.

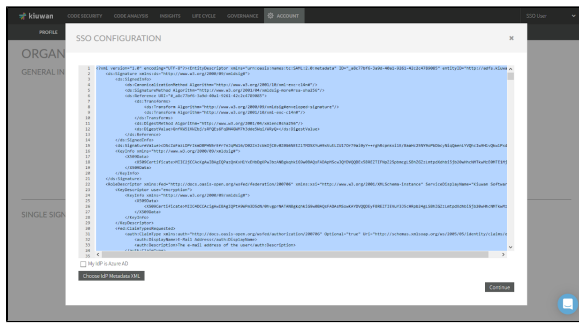
- Once SSO is activated, it is NOT possible to disable it or re-migrate users to the previous Kiuwan domain.
- Even though the activation process is completed, you will need to register Kiuwan as SP in your IdP. Till then, you can not use SSO. See section on "Kiuwan's metadata configuration in ADFS"

Click **Continue** to upload your IdP Metadata XML.

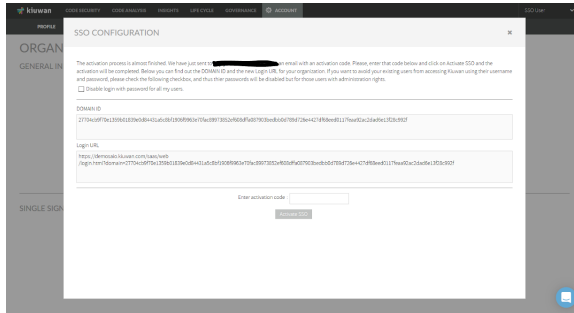
In a typical ADFS installation, you can commonly get it at [https://<your\\_idp\\_domainname>/FederationMetadata/2007-06/FederationMetadata.xml](https://<your_idp_domainname>/FederationMetadata/2007-06/FederationMetadata.xml)



If your IdP is **Azure AD**, check the checkbox **My IdP is Azure AD**.



Once it's loaded, click **Continue**.



At this moment, you should have **received an email with an activation code** as well as Domain Id and Login URL. Enter the activation code and click **Activate SSO** button.



- If you want to **avoid currently existing Kiuwan users to login using former credentials** (username and password), check **Disable login with password** for all my users. By checking this option, all the users will be forced to log in through SSO (using the provided URL).
- If you don't check that option, existing users can still log in using user/password, but using the new URL. The older Kiuwan URL will not work anymore because all the users have been migrated to SSO.

**IMPORTANT:** If you have users who use the Kiuwan Local Analyzer Checking this option, a user launching the Kiuwan Local Analyzer will not be able to use it unless:

- he configures KLA to use SSO, selecting "Enable Single sign-on" and filling the Domain ID and connection credentials, or
- an administrator allows him to still use kiuwan credentials (see [User Management](#)) AND the KLA is configured filling the Domain ID and with "Enable Single sign-on" unchecked.

Admin users can ALWAYS login both ways. And also, can always modify which Kiuwan users are allowed to login using Kiuwan credentials (see [User Management](#)).

Example mail with activation code:

To continue with SSO activation process, please enter the following activation code in the screen:

Activation Code: **aQ1JH15YBR**

Remember that after SSO activation, you have to use the following link to log into Kiuwan using your username and password:

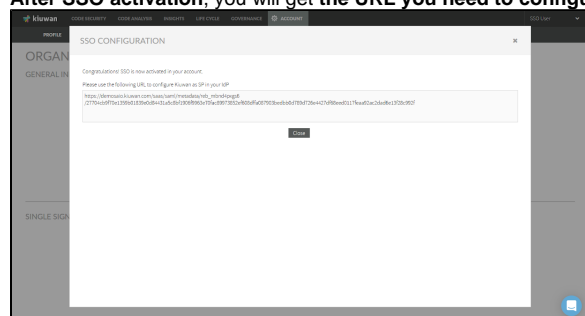
Login URL: <https://demoaio.kiuwan.com/saas/web/login.html?domain=27704cb9f70e1359b01839e0d84431a5c8bf1906f9963e70fac89973852ef608dffa087903bedbb0d789d726e4427df68eed0117f6a92ac2dad6e13f28c992f>

And this is your DOMAIN ID, necessary to use the Kiuwan Local Analyzer, Kiuwan REST API, and Kiuwan plugins:

DOMAIN ID: 27704cb9f70e1359b01839e0d84431a5c8bf1906f9963e70fac89973852ef608dffa087903bedbb0d789d726e4427df68eed0117f6a92ac2dad6e13f28c992f

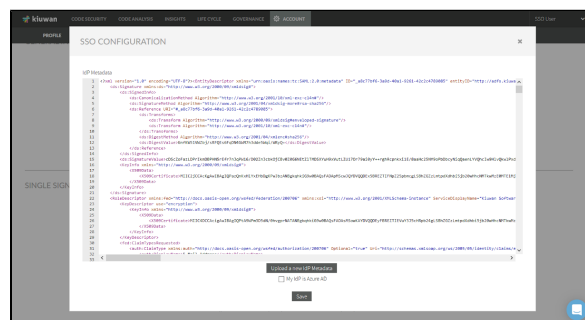
Thanks for using Kiuwan. Have fun analyzing!  
The Kiuwan Team

**After SSO activation, you will get the URL you need to configure Kiuwan as an SP in your IdP.**

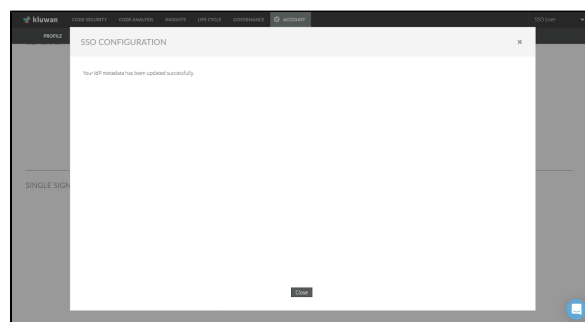


Close the page and **the Kiuwan SSO configuration is done!**

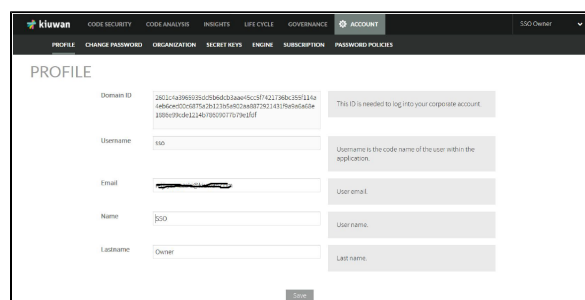
If you need to update existing metadata with new IdP metadata, go to the SSO initial configuration page and click **Upload a new IdP Metadata**.



Click **Save** to complete the update



**After metadata configuration, go to Account Management > Profile and you will see the following data in your Kiuwan account.**



**Domain ID** only appears when your Kiuwan account is configured to use SSO.

- This ID is needed to login to your Kiuwan account. It is shared by all users of a Kiuwan account, but unique for every Kiuwan account.

**Username** field contains your Kiuwan username. It matches the Claim mapping (Name ID) defined in your IdP when you defined Kiuwan as Service Provider (see image above for ADFS).

Email, Name and Lastname fields are descriptive data about the user.

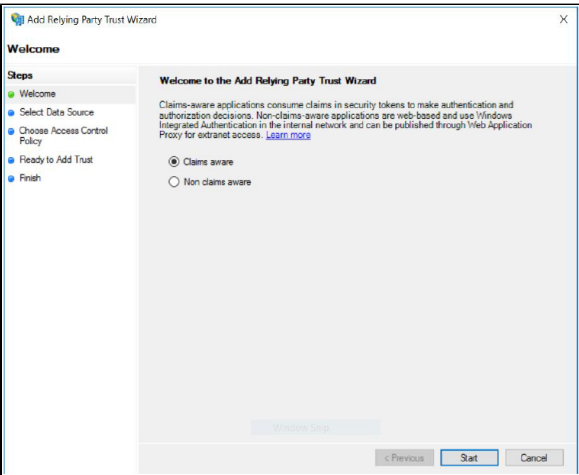
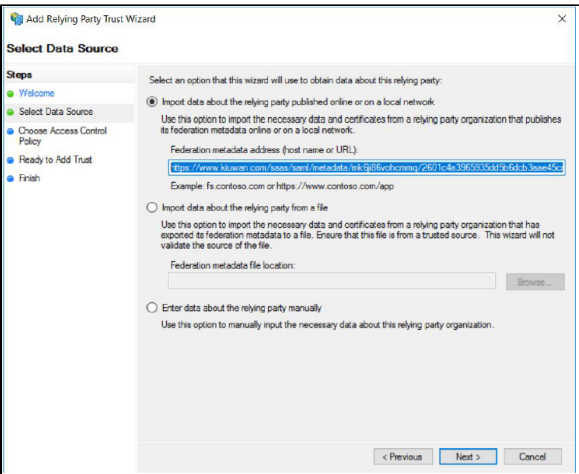
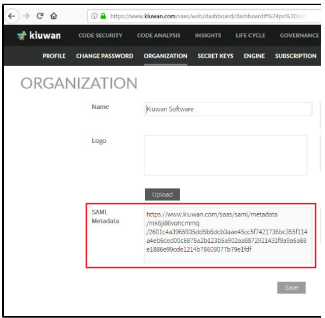
## IdP configuration: How to configure Kiuwan as Service Provider

The IdP (Identity Provider) must be configured to recognize Kiuwan as an SP (Service Provider).

Any SAML-compliant IdP (Active Directory FS, Azure AD, CA Single Sign-On, etc) follows its configuration method, although steps are similar.

We provide a **detailed example of how to configure Active Directory Federation Services (ADFS)**. For other IdPs please refer to your sysadmins or product documentation.

## Active Directory Federation Services (ADFS) configuration

	<ol style="list-style-type: none"><li>1. Open <b>ADFS's Add Relying Pa</b></li><li>2. Select <b>Claims aware</b> and click</li></ol>
	<ol style="list-style-type: none"><li>3. Then, ADFS will ask you about <b>Ki</b></li></ol> <p>Ideally, if your ADFS can reach Kiuw</p> <p>Then you must <b>provide the address</b>: <b>Organization</b> page (see image belo</p> 
	<p>If your ADFS cannot reach the Kiuw from a file.</p> <p>In this case, you must previously do in a browser that can access the Kiu</p>

Add Relying Party Trust Wizard

**Specify Display Name**

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

Display name:

Notes:

< Previous Next > Cancel

4. Provide a **Display name** for Kiwan  
(It doesn't have to be a domain host

Add Relying Party Trust Wizard

**Choose Access Control Policy**

Choose an access control policy:

Name	Description
Permit everyone	Grant access to everyone
Permit everyone and require MFA	Grant access to everyone and require MFA
Permit everyone and require MFA for specific group	Grant access to everyone and require MFA for specific group
Permit everyone and require MFA from extranet access	Grant access to the intranet users and require MFA from extranet access
Permit everyone and require MFA from unauthenticated devices	Grant access to everyone and require MFA from unauthenticated devices
Permit everyone and require MFA, allow automatic device registration	Grant access to everyone and require MFA, allow automatic device registration
Permit everyone for intranet access	Grant access to the intranet users
Permit everyone from the Internet	Grant access to everyone from the Internet

Policy:  
Permit everyone

☐ I do not want to configure access control policies at this time. No user will be permitted access for this application.

< Previous Next > Cancel

5. Choose the **Access Control Policy**  
6. Click **Next** to confirm.

Add Relying Party Trust Wizard

**Ready to Add Trust**

The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.

Monitoring Identifiers Encryption Signature Accepted Claims Organization Endpoints Notifications

Specify the monitoring settings for this relying party trust.

Relying party's federation metadata URL:

☒ Monitor relying party

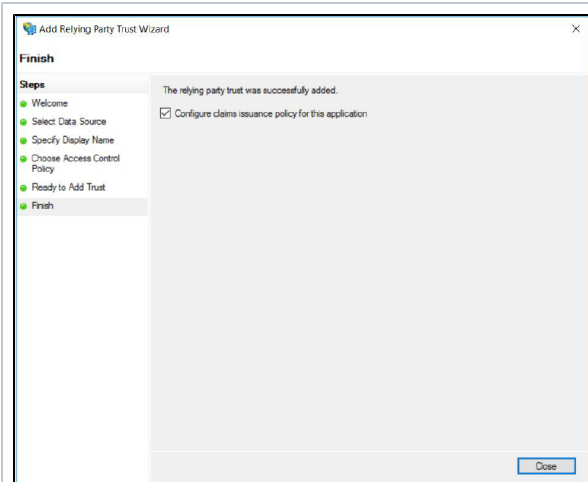
☒ Automatically update relying party

This relying party's federation metadata data was last checked on:  
25/03/2019

This relying party was last updated from federation metadata on:  
25/03/2019

< Previous Next > Cancel

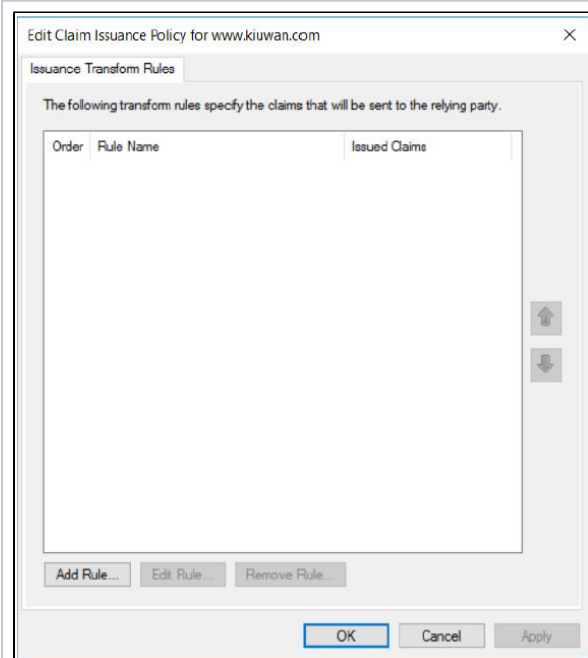
7. Review the information from the S



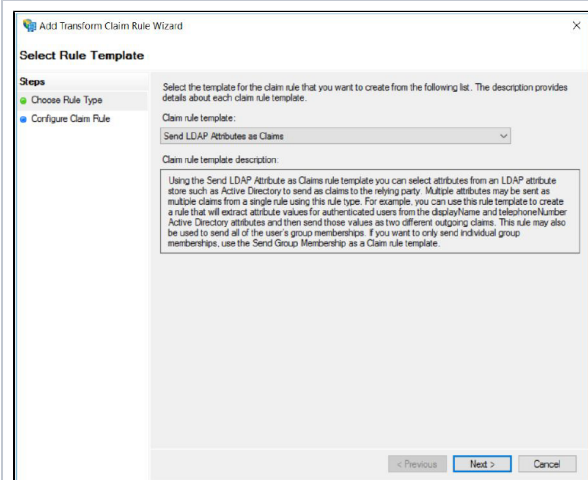
Notice that **Configure claims issuance**

When checked, you will define how t

8. Click **Close** and **Edit Claim Issua**



9. Click **Add Rule** to open **Add Trar**



10. **Select the template rule** most a

In the example, we select to map an

**Add Transform Claim Rule Wizard**

**Configure Rule**

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name: Email as NameID

Rule template: Send LDAP Attributes as Claims

Attribute store: Active Directory

Mapping of LDAP attributes to outgoing claim types:

LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
E-Mail Addresses	Name ID

< Previous Finish Cancel

You can select whatever LDAP attrit that attribute to the Name ID claim

Do not select any other claim type

Kiuwan will store as a username the

11. Click **Finish**.

**Edit Claim Issuance Policy for www.kiuwan.com**

**Issuance Transform Rules**

The following transform rules specify the claims that will be sent to the relying party.

Order	Rule Name	Issued Claims
1	Email as NameID	Name ID

Add Rule... Edit Rule... Remove Rule...

OK Cancel Apply

12. Click **Apply** to apply changes.

## How to log into Kiuwan in a Web SSO scenario

**i** The first time you log in *at Kiuwan in SSO-mode, you need to specify the full URL* such as:

<https://www.kiuwan.com/saas/web/login.html?sso=on&domain=2601c4a3965935dd5b6dcb3aae45cc5f7421736bc355f114a4eb6ced00c6875a2b123b5a902aa8872921431f9a9a6a68e1886e99cde1214b78609077b79e1fdf>

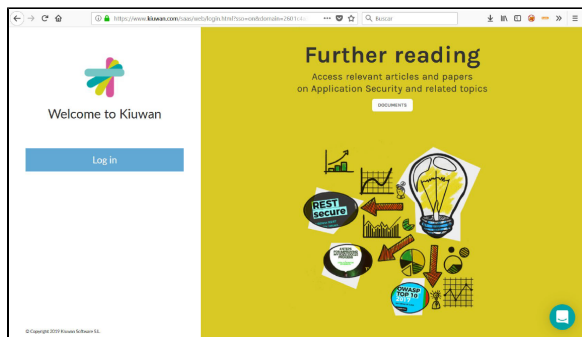
Please note that, once SSO has been activated, the login URL must specify both **SSO** and **do main** parameters.

- **sso=on** will make Kiuwan authenticate against the configured IdP
- **sso=off** will make Kiuwan authenticate locally, so login page will ask for credentials and will check them against kiuwan database (obviously this process will only work for users that are allowed to log in with kiuwan passwords, see [SSO login vs username-passwordlogin](#))

If you don't specify SSO, it defaults to **off**.

Most commonly, in an SSO environment, you will access Kiuwan from an existing link in a corporate intranet page, so the Kiuwan URL should be changed to it and you will not need to type the URL manually. Regardless, once you have successfully accessed Kiuwan for the first time, **your browser will store the domain ID**, so you can just type <https://www.kiuwan.com> and everything will work.

Then, the Kiuwan SSO Login page will be displayed.

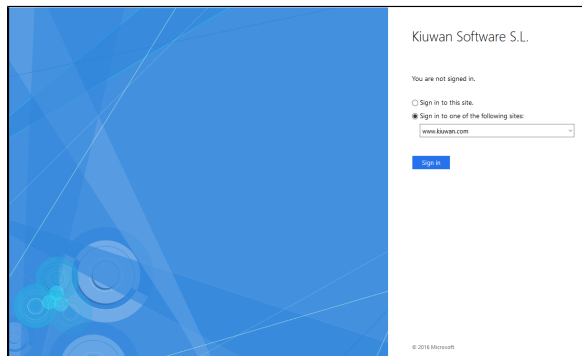


Just click **Log In** and the SSO-SAML protocol will be activated.

- If you were already successfully authenticated, you will log in to Kiuwan.
- If not, you will be redirected to your organizational authentication page. Once authenticated, you will be redirected to the Kiuwan dashboard.

**An alternative method to login to Kiuwan is from your IdP.**

If you are using ADF, you will find a URL like this: [https://<your\\_idp\\_hostname>/adfs/ls/idpInitiatedsignon.htm](https://<your_idp_hostname>/adfs/ls/idpInitiatedsignon.htm)



Just select the site (the Display Name defined at your IdP). Provide your credentials to be redirected to the Kiuwan dashboard.

## How to configure Kiuwan clients to work with SSO - SAML

**After configuring SSO, your web users can immediately log in to the Kiuwan website using the new login URL.**

But **Kiuwan “clients”** (i.e. Kiuwan Local Analyzer, Kiuwan 4 Developers, and any custom program using Kiuwan REST-API) **need to be configured to use SSO.**

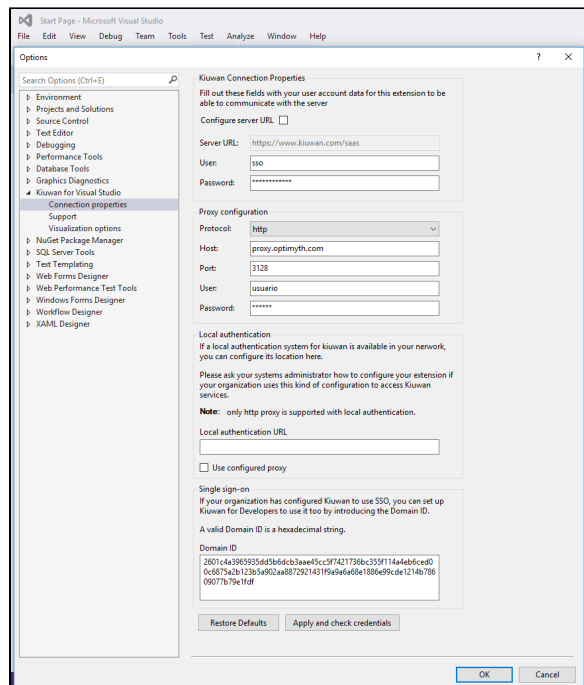
### Kiuwan Local Analyzer (KLA): SSO configuration

Please refer to [Single Sign On](#) for more information on how to configure Kiuwan Local Analyzer with SSO.

### Kiuwan for Developers (K4D): SSO configuration

K4D needs to be configured with the Domain ID of your account.

Go to your **IDE's Kiuwan configuration**, select **Connection Properties > Single Sign-On** and enter your **Domain ID**.



## REST-API: SSO configuration

For custom programs using Kiuwan **REST-API calls**, you have to add a **new header (X-KW-CORPORATE-DOMAIN-ID)** to indicate the Domain ID to pass the BASIC authentication.

For example:

```
curl -H "X-KW-CORPORATE-DOMAIN-ID: {domain.id}" -u {username}:{password} https://api.kiuwan.com/info
```



To use REST-API on customers with Single Sign-On, the user must be authorized by the administrator to continue using Kiuwan credentials. In this case, the user must authenticate not only providing their username and password in Kiuwan but also indicating the domain to which they belong to.

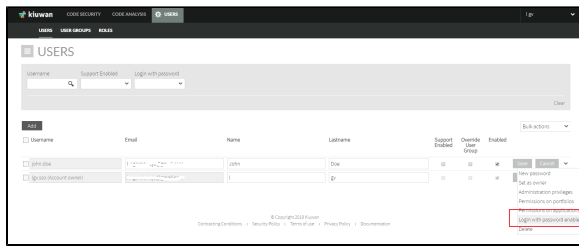
## SSO login vs username-password login

When a Kiuwan account is converted to SSO-enabled, by default:

1. **All existing users must use the new login URL** (see [How to login at Kiuwan in a Web SSO scenario](#))
  - a. Previous URL login (<https://www.kiuwan.com/saas/web/login.html>) will not work anymore
2. **Usernames and permissions are entirely preserved**
  - a. Only the authentication mechanism has changed. Usernames, assigned roles, permissions, user groups, etc are maintained.
3. **Existing users (not admins) are not allowed to log in to Kiuwan using former Kiuwan password**
  - a. They will be authenticated by the configured identity provider (IdP), not by Kiuwan.

Nevertheless, you might want **certain users to continue to be authenticated by Kiuwan**, i.e, some user might choose to authenticate either by SSO or by Kiuwan.

The Kiuwan admin can enable username/password access through the **User Administration** page, enabling **Login with password enabled to** selected users



Users with privilege **Login with password enabled** can then login to Kiuwan in two ways:

1. Authenticated by SSO
  - a. [https://www.kiuwan.com/saas/web/login.html?sso=on&domain=<my\\_domain\\_id>](https://www.kiuwan.com/saas/web/login.html?sso=on&domain=<my_domain_id>)
2. Authenticated by Kiuwan (by password)
  - a. [https://www.kiuwan.com/saas/web/login.html?sso=off&domain=<my\\_domain\\_id>](https://www.kiuwan.com/saas/web/login.html?sso=off&domain=<my_domain_id>)

## Adding a new user in an SSO-enabled account

In an SSO-enabled account, when you create a new user, you can decide if that user can access Kiuwan with a password (besides SSO).

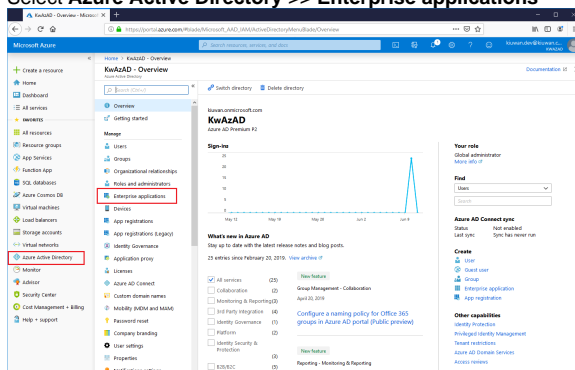
Just check the **Enable login with password** option in the **New User** page and click on **Generate password** to let him/her know.

## Appendix - Azure Active Directory configuration: How to configure Kiuwan as Service Provider

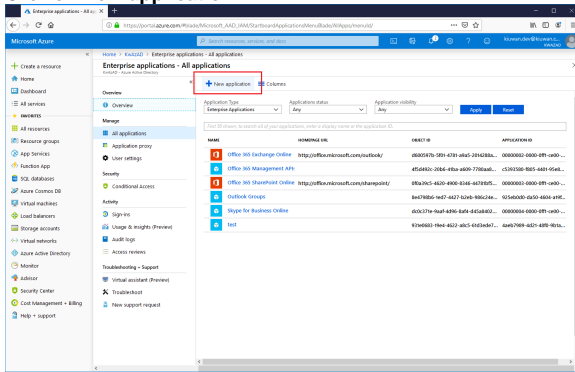
You must configure your Idp (Azure AD) so it recognizes Kiuwan as an SP (Service Provider).

In Azure AD, create an **Enterprise Application** (Kiuwan SSO, in this example).

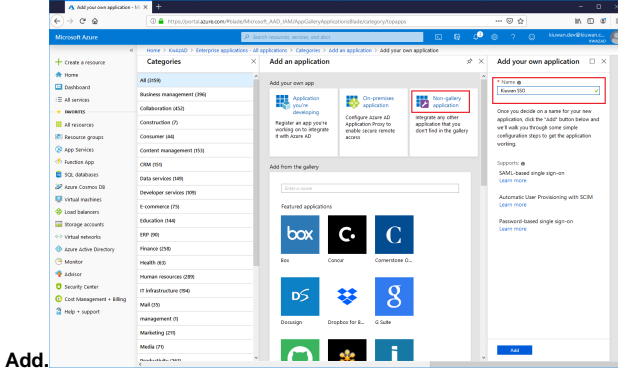
### 1. Select Azure Active Directory >> Enterprise applications



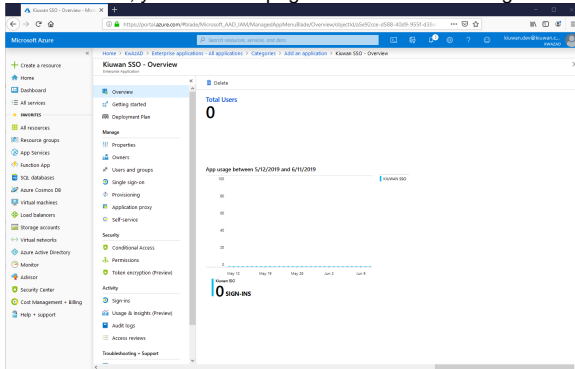
## 2. Click on New application.



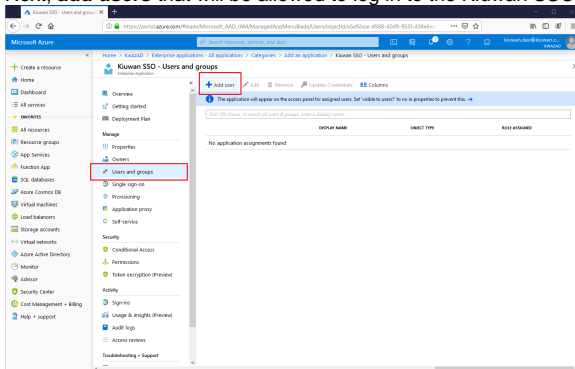
## 3. Select **Non-gallery application** and fill in the app name (Kiuwan SSO in our example) and click



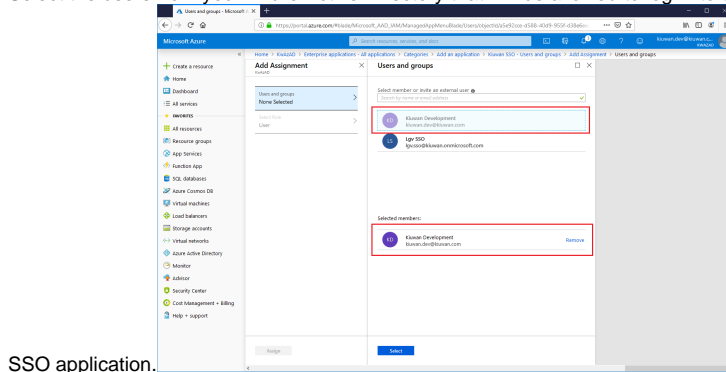
## 4. When created, you will see a page similar to the following:



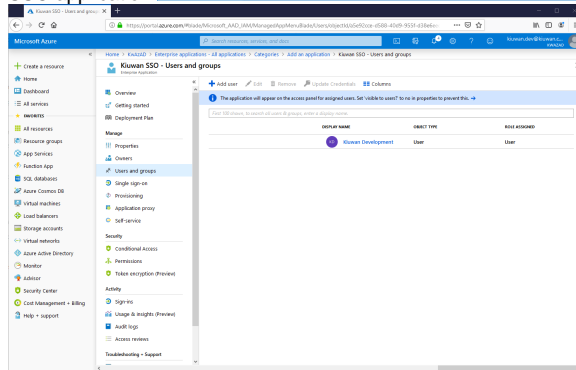
## 5. Next, **add users** that will be allowed to log in to the Kiuwan SSO application.



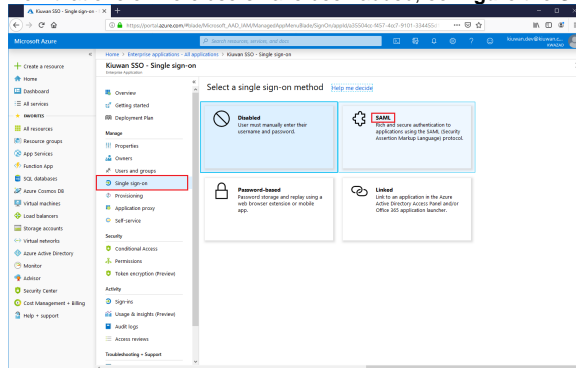
6. Select the users from your Azure Active Directory that will be allowed to log in to the Kiwanan



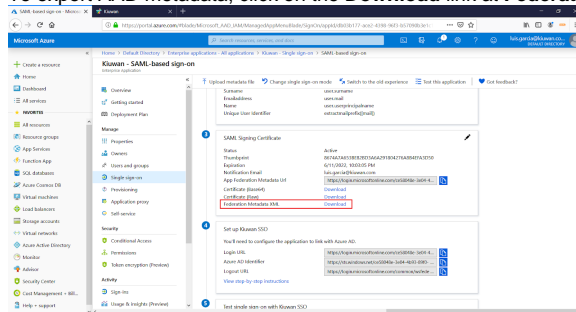
SSO application.



7. Now that one or more users have been added, configure the Single sign-on.



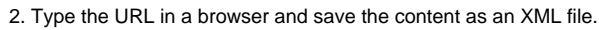
8. Export the Azure Active Directory metadata and import it to Kiwanan.  
To export AAD metadata, click on the Download link at Federation Metadata XML.

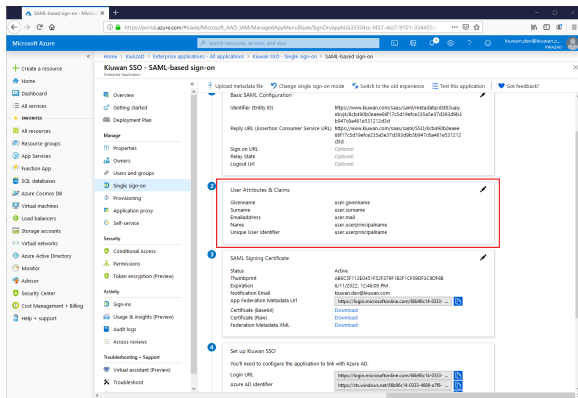


The downloaded XML file needs to be imported into your Kiwanan account, as shown before.

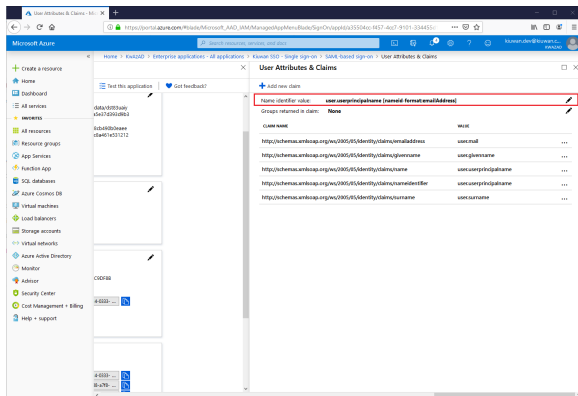
After importing AAD metadata into Kiwanan, your Kiwanan account will be ready to generate its metadata that you will import into AAD.

1. To export Kiwanan metadata, go to **Account Management > Organization** and you will see the URL to download Kiwanan metadata.



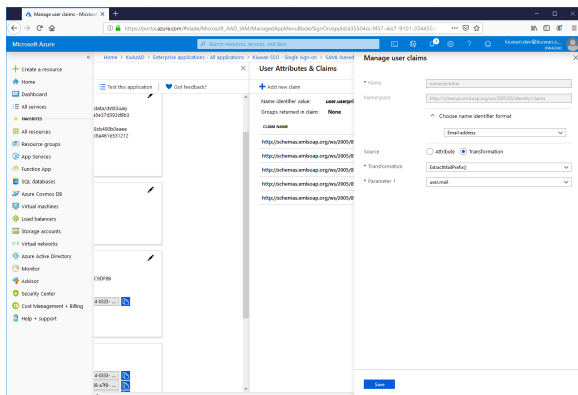


4. Select the **Name identifier value** and set up the policy on how to manage your ADA usernames to Kiuwan usernames.

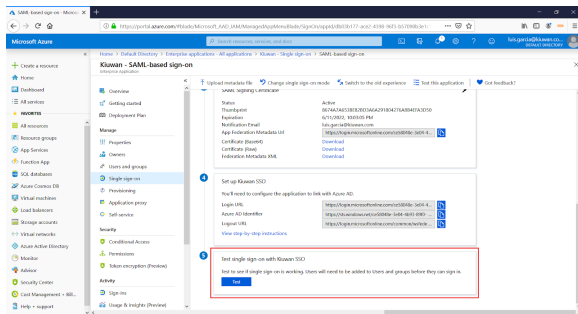


5. In this example, we take the first part of the email.

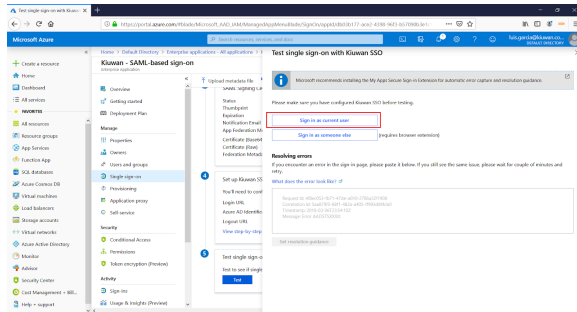
For example, an AAD user with email **john.doe@domain.com** will be mapped to john.doe when sent to Kiuwan.



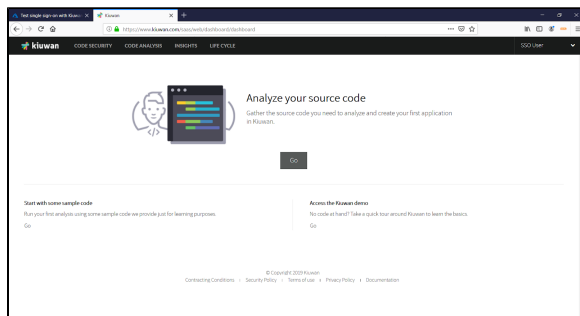
6. Now, click **Test** to test Single Sign-On with the Kiuwan SSO app.



7. Select the user (the current one or someone else)



8. Because you are already logged in ADD (and therefore authenticated) **you will be forwarded directly to the Kiuwan app.**



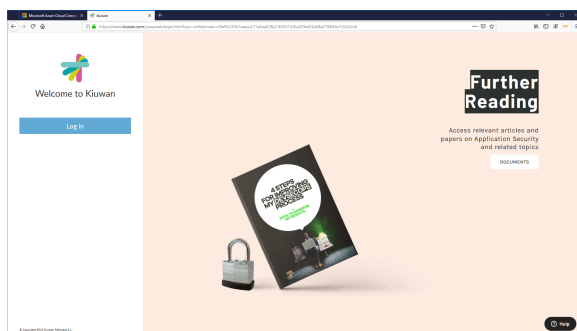
9. Login from the Kiuwan site

## Login from the Kiuwan site

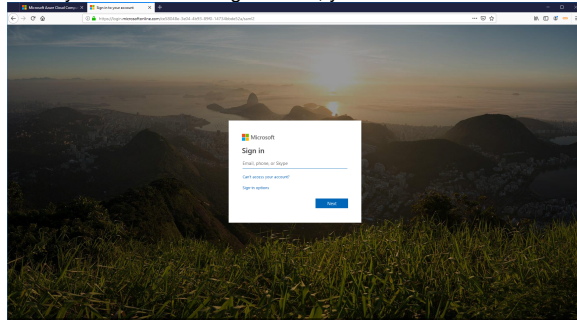
To log in from the Kiuwan site, you must go to SSO URL (remember to set sso=on and set the domain)

For example <https://www.kiuwan.com/saas/web/login.html?sso=on&domain=f0e9c230b1caecc231a6aa636a7400315c6a02fac03a68a01864bc7c5452a5>

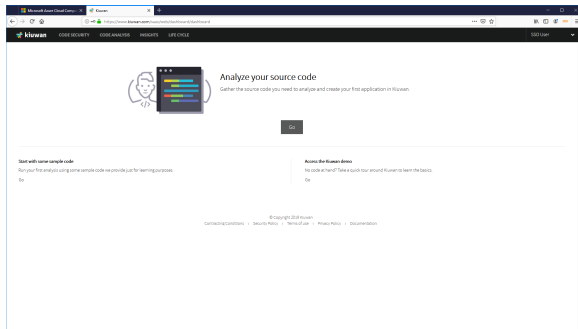
You will be presented with the login page (without need to write your credentials)



When you click on the Login button, you will be forwarded to the Azure login page:



Type your credentials (AAD will authenticate you), and (if successful) you will be logged in at Kiuwan site



You need to authenticate even if you are logged in at AAD, because the second authentication has been forced by Kiuwan. Very often IdPs (AAD, ADFS, etc) send to Kiuwan *old* auth tokens, making SSO fail. To prevent these situations, **Kiuwan forces IdP to perform the auth process** and send to Kiuwan a *fresh* token.