RCDEVS IDENTITY PROVIDER AND INTEGRATIONS

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs Security.

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

RCDevs Identity Provider and integrations

SSO Federation SAML OpenID Nextcloud Guacamole Grafana GitLab OnlyOffice Identity Provider Service Provider IDP SP

1. Overview

This document will present you how to use WebADM as Identity Provider (IDP) with different Service Provider (SP) which will consume OpenOTP for authentication processes. We will also see how we can configure and return different information per service provider through users/groups and client policies.

The installation of `OpenID/SAML` IdP is straightforward and only consists of running the self-installer or install the `openid` package from RCDevs repositories and configure the application in WebADM.

You do not have to modify any files in the OpenID install directory! The web application configurations are managed and stored in the LDAP configured with by WebADM. To configure OpenID/SAML provider, your must login on WebADM as super administrator and go to the `Applications` menu. Click `CONFIGURE` on OpenID/SAML to enter the web-based configuration.

OpenID/SAML application logs are accessible in the `Databases` menu in WebADM.

Note: To be able to use OpenID/SAML, any LDAP users' accounts must be a activated in WebADM.

You can embed the `SAML & OpenID` Webapp on your website in an HTML iFrame or Object.

```
#Example
<object data="https://<webadm_addr>/webapps/openid?inline=1" />
```

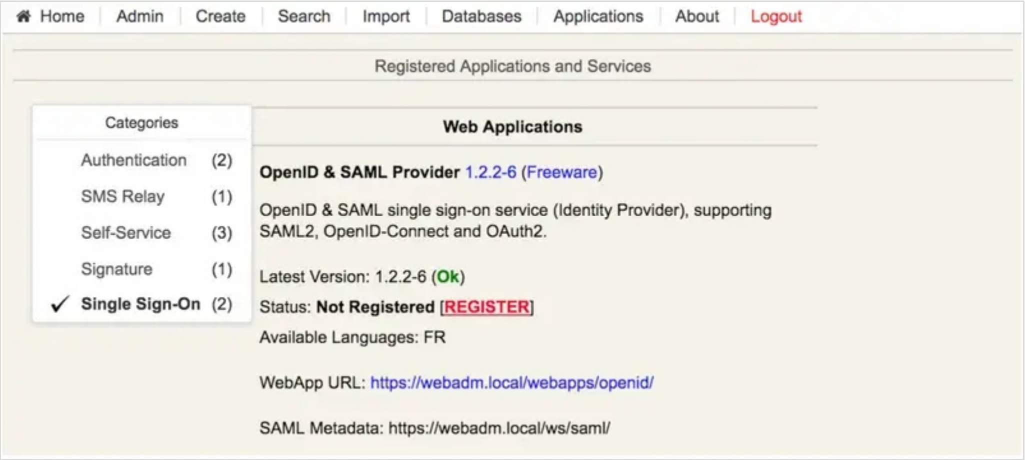
Once your IDP global configuration is done, the best practice is to create `Client policy` for each Service Provider you are configuring with your IDP. That will be describe later in that documentation.

2. WebADM IDP configuration

First, we need a WebADM server with *MFA Authentication Server* and *OpenID & SAML Provider* packages installaled.

Once the server is up and running, we can configure it as a SAML Identity Provider (IdP).

Login to the `WebADM Admin Portal` and navigate to `Applications` tab > `Singe Sign-On` > `OpenID & SAML Provider`. Click then `REGISTER` button. The LDAP object containing the IDP configuration is created.



Registered Applications and Services

Categories	
Authentication	(2)
SMS Relay	(1)
Self-Service	(3)
Signature	(1)
✓ Single Sign-On	(2)

Web Applications

OpenID & SAML Provider 1.2.2-6 (Freeware)

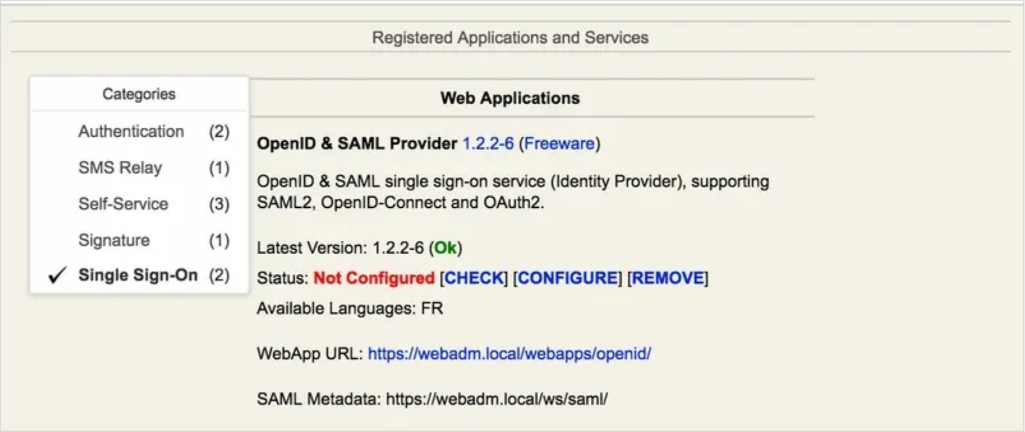
OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.2.2-6 (Ok)
Status: **Not Registered** [REGISTER]
Available Languages: FR

WebApp URL: https://webadm.local/webapps/openid/

SAML Metadata: https://webadm.local/ws/saml/

Once the appliaction is registered, click on `CONFIGURE` button to configure the IDP:



Registered Applications and Services

Categories	
Authentication	(2)
SMS Relay	(1)
Self-Service	(3)
Signature	(1)
✓ Single Sign-On	(2)

Web Applications

OpenID & SAML Provider 1.2.2-6 (Freeware)

OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

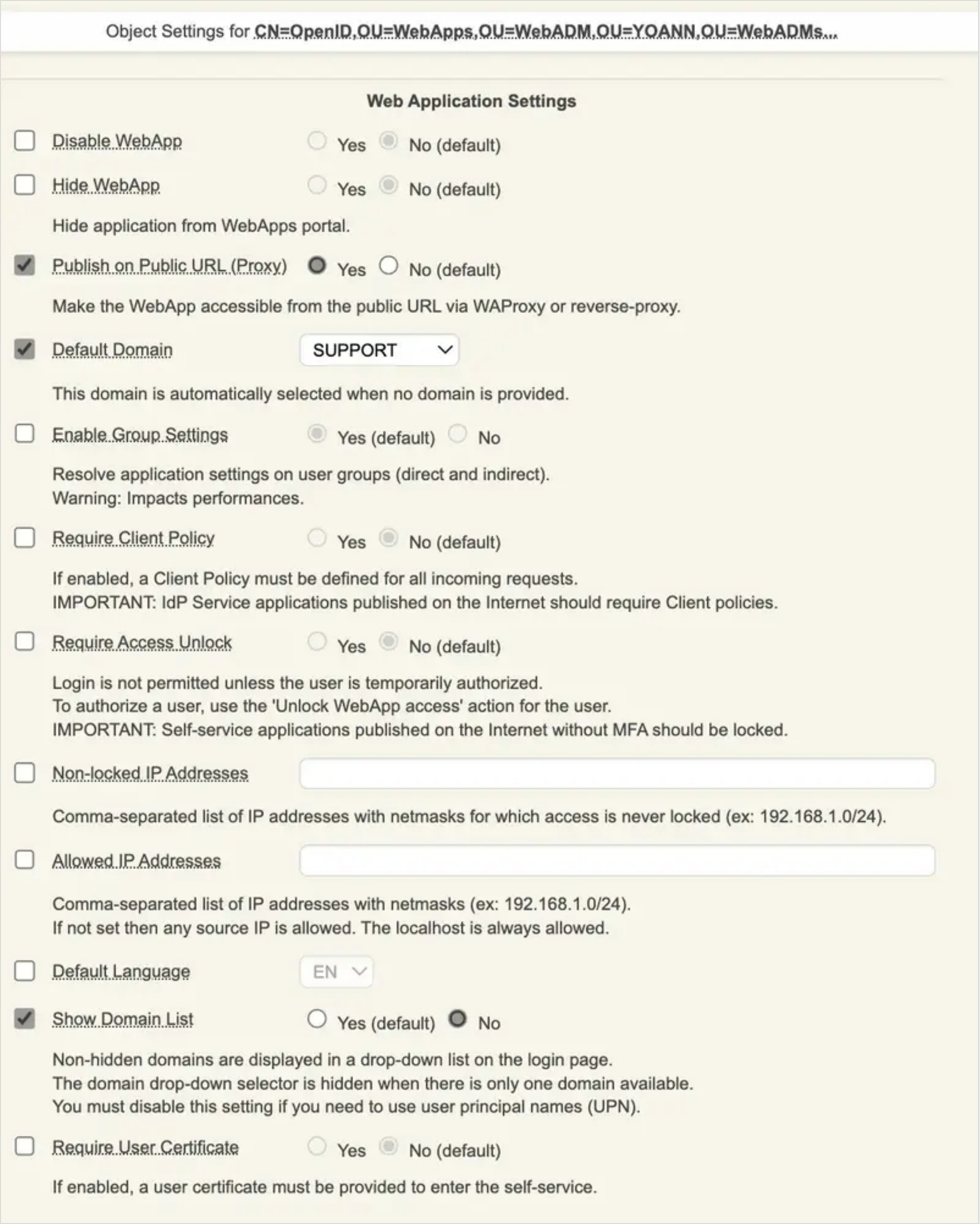
Latest Version: 1.2.2-6 (Ok)
Status: **Not Configured** [CHECK] [CONFIGURE] [REMOVE]
Available Languages: FR

WebApp URL: https://webadm.local/webapps/openid/

SAML Metadata: https://webadm.local/ws/saml/

2.1 Web Application Settings and Common Features

You are now in the global configuration of your `OpenID & SAML Identity Provider`.



Object Settings for CN=OpenID,OU=WebApps,OU=WebADM,OU=YOANN,OU=WebADMs...

Web Application Settings

☐ Disable WebApp ○ Yes ● No (default)

☐ Hide WebApp ○ Yes ● No (default)

Hide application from WebApps portal.

☑ Publish on Public URL (Proxy) ● Yes ○ No (default)

Make the WebApp accessible from the public URL via WAProxy or reverse-proxy.

☑ Default Domain SUPPORT

This domain is automatically selected when no domain is provided.

☐ Enable Group Settings ● Yes (default) ○ No

Resolve application settings on user groups (direct and indirect).
Warning: Impacts performances.

☐ Require Client Policy ○ Yes ● No (default)

If enabled, a Client Policy must be defined for all incoming requests.
IMPORTANT: IdP Service applications published on the Internet should require Client policies.

☐ Require Access Unlock ○ Yes ● No (default)

Login is not permitted unless the user is temporarily authorized.
To authorize a user, use the 'Unlock WebApp access' action for the user.
IMPORTANT: Self-service applications published on the Internet without MFA should be locked.

☐ Non-locked IP Addresses

Comma-separated list of IP addresses with netmasks for which access is never locked (ex: 192.168.1.0/24).

☐ Allowed IP Addresses

Comma-separated list of IP addresses with netmasks (ex: 192.168.1.0/24).
If not set then any source IP is allowed. The localhost is always allowed.

☐ Default Language EN

☑ Show Domain List ○ Yes (default) ● No

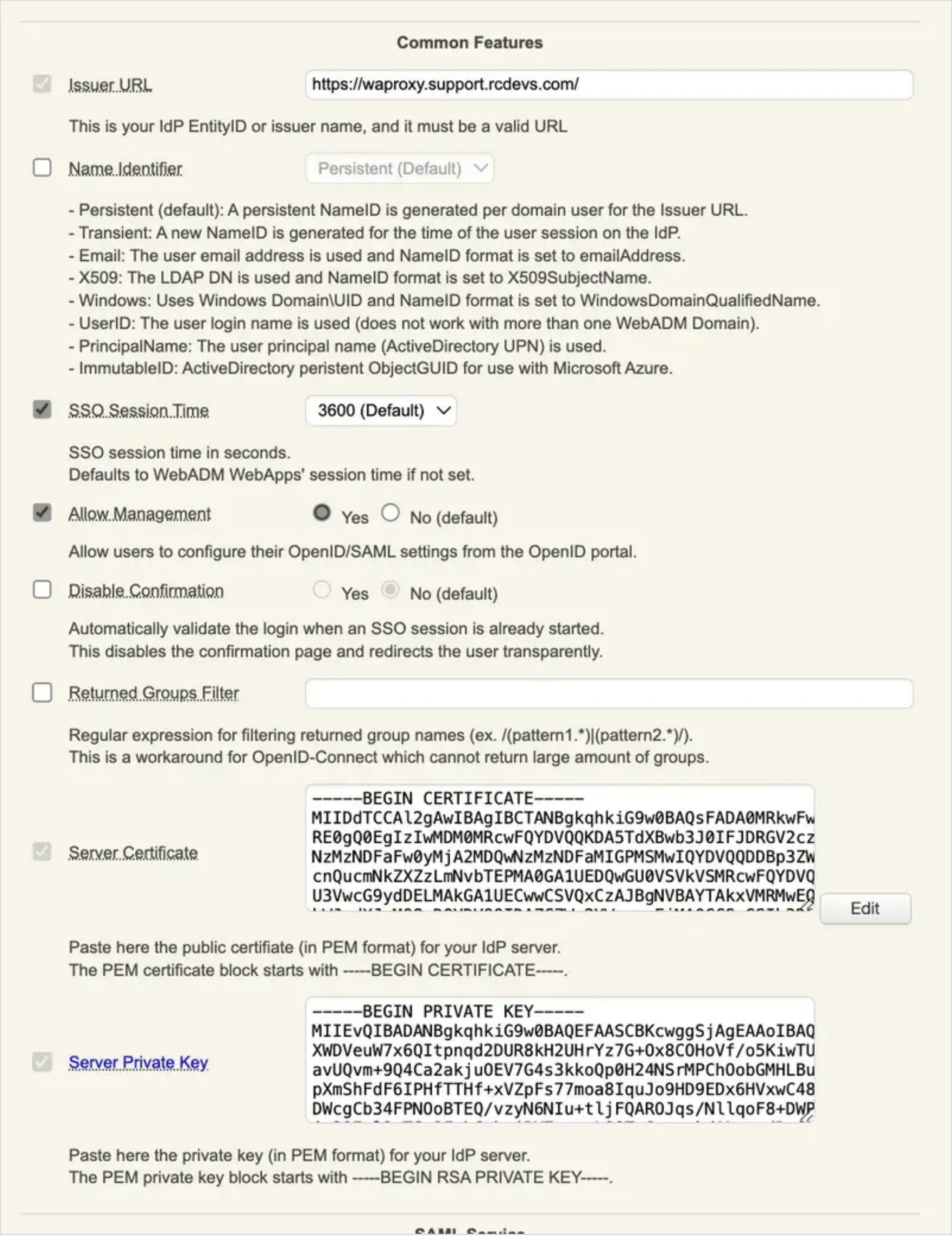
Non-hidden domains are displayed in a drop-down list on the login page.
The domain drop-down selector is hidden when there is only one domain available.
You must disable this setting if you need to use user principal names (UPN).

☐ Require User Certificate ○ Yes ● No (default)

If enabled, a user certificate must be provided to enter the self-service.

Configure the setting you would like to apply. On my side, I published the Web application on my WAProxy, hidden the `Domain List` because multiple domains are available on my infrastructure and I do not want that information displayed on my IDP login page. I also enforced a default domain but remember that this can be configured at the `Client Policy` level.

We are now entering in the `Common Features` section.



Common Features

☑ Issuer URL: https://waproxy.support.rcdevs.com/

This is your IdP EntityID or issuer name, and it must be a valid URL

☐ Name Identifier: Persistent (Default)

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- PrincipalName: The user principal name (ActiveDirectory UPN) is used.
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

☑ SSO Session Time: 3600 (Default)

SSO session time in seconds.
Defaults to WebADM WebApps' session time if not set.

☑ Allow Management: ● Yes ○ No (default)

Allow users to configure their OpenID/SAML settings from the OpenID portal.

☐ Disable Confirmation: ○ Yes ● No (default)

Automatically validate the login when an SSO session is already started.
This disables the confirmation page and redirects the user transparently.

☐ Returned Groups Filter

Regular expression for filtering returned group names (ex. /(pattern1.*)|(pattern2.*)/).
This is a workaround for OpenID-Connect which cannot return large amount of groups.

☑ Server Certificate

```
-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgIBCTANBgkqhkiG9w0BAQsFADA0MRkwFw
RE0gQ0EgIzIwMDM0MRcwFQYDVQQKDA5TdXBwb3J0IFJDRGV2cz
NzMzNDFaFw0yMjA2MDQwNzMzNDFaMIGPMSMwIQYDVQQDDBp3ZW
cnQucmNkZXZzLmNvbTEPMA0GA1UEDQwGU0VSVkVSMRcwFQYDVQ
U3VwcG9ydDELMAkGA1UECwwCSVQxCzAJBgNVBAYTAkxVMRMwEQ
```

Paste here the public certifiate (in PEM format) for your IdP server.
The PEM certificate block starts with -----BEGIN CERTIFICATE-----.

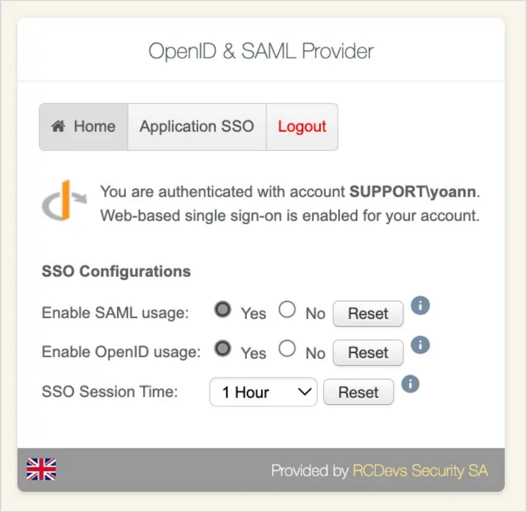
☑ Server Private Key

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQ
XWDVeuW7x6QItpnqd2DUR8kH2UHrYz7G+Ox8COHoVf/o5KiwTU
avUQvm+9Q4Ca2akjuOEV7G4s3kkoQp0H24NSrMPChOobGMHLBu
pXmShFdF6IPHfTTHf+xVZpFs77moa8IquJo9HD9EDx6HVxwC48
DWcgCb34FPNOoBTEQ/vzyN6NIu+tljFQAROJqs/NllqoF8+DWP
```

Paste here the private key (in PEM format) for your IdP server.
The PEM private key block starts with -----BEGIN RSA PRIVATE KEY-----.

- The `Issuer URL` or `EntityID` is a unique identifier that is used to identify a specific entity in the SAML authentication and authorization protocol. A SAML entity ID is typically a URL or URI that is assigned to the entity, and it is used to identify the entity in SAML messages and metadata. That setting will refer to `Issuer` value for OpenID. In that documentation, I configured my `Issuer URL` with the public DNS name targeting my WebADM infrastructure. In most of the case, the IDP URL will be a public URL which can be easily proxied with WebADM Publishing Proxy or with another Reverse Proxy solution.
- The `Name Identifier` setting is the unique identifier of the user. It should be non-volatile and opaque. It should not contain personal information or information that is changeable over time, such as the user's name or email address. The accepted `Name Identifier` may vary according to the Service Provider you are integrating and for that reason it can make more sens to configure it per Service Provider `Client Policy`.
- The `SSO Session Time` define the time for a user session remains valid on the IDP.
- The `Allow Management` setting provides the possibility to your end-users to enable/disable the SAML/OpenID usage for their account and configure their SSO Session timeout. It is recommanded to disabled that setting by default. Example below

of end-user view once authenticated on the IDP and when that setting is enabled:



OpenID & SAML Provider

You are authenticated with account **SUPPORT\yoann**.
Web-based single sign-on is enabled for your account.

SSO Configurations

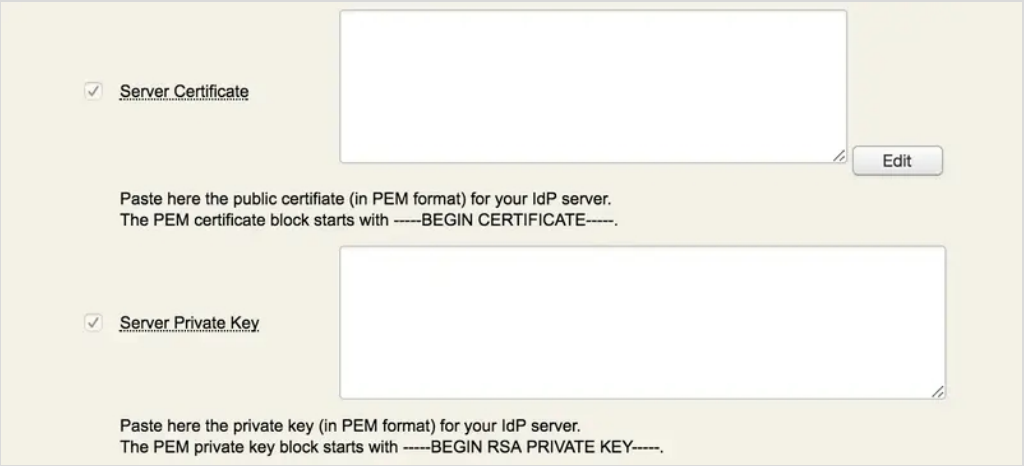
Enable SAML usage: ● Yes ○ No Reset

Enable OpenID usage: ● Yes ○ No Reset

SSO Session Time: 1 Hour Reset

Provided by RCDevs Security SA

- The `SSO Session Time` setting allow the transparent redirection to an Service Provider once the user is authenticated.
- The `Returned Groups Filter` is a regular expression which can be configured in order to filter groups returned in the SAML or OpenID responses based on the RegEx match.
- The `Server Certificate` and `Server Private Key` settings are mandatory and will be used for request signing purposes. Click `Edit` and `Generate` buttons, then a certificate with WebADM internal PKI is issued.

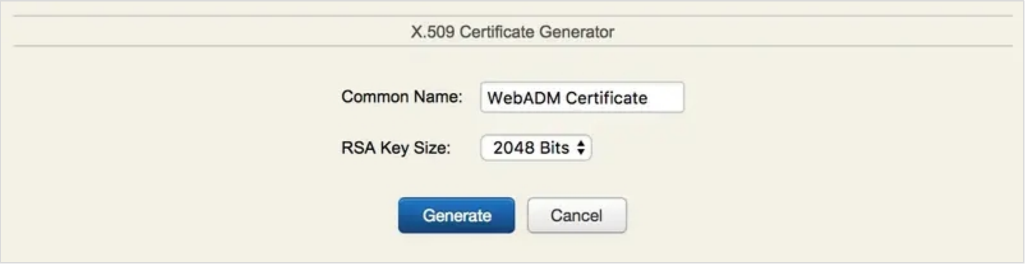


☑ Server Certificate

Paste here the public certifiate (in PEM format) for your IdP server.
The PEM certificate block starts with -----BEGIN CERTIFICATE-----.

☑ Server Private Key

Paste here the private key (in PEM format) for your IdP server.
The PEM private key block starts with -----BEGIN RSA PRIVATE KEY-----.

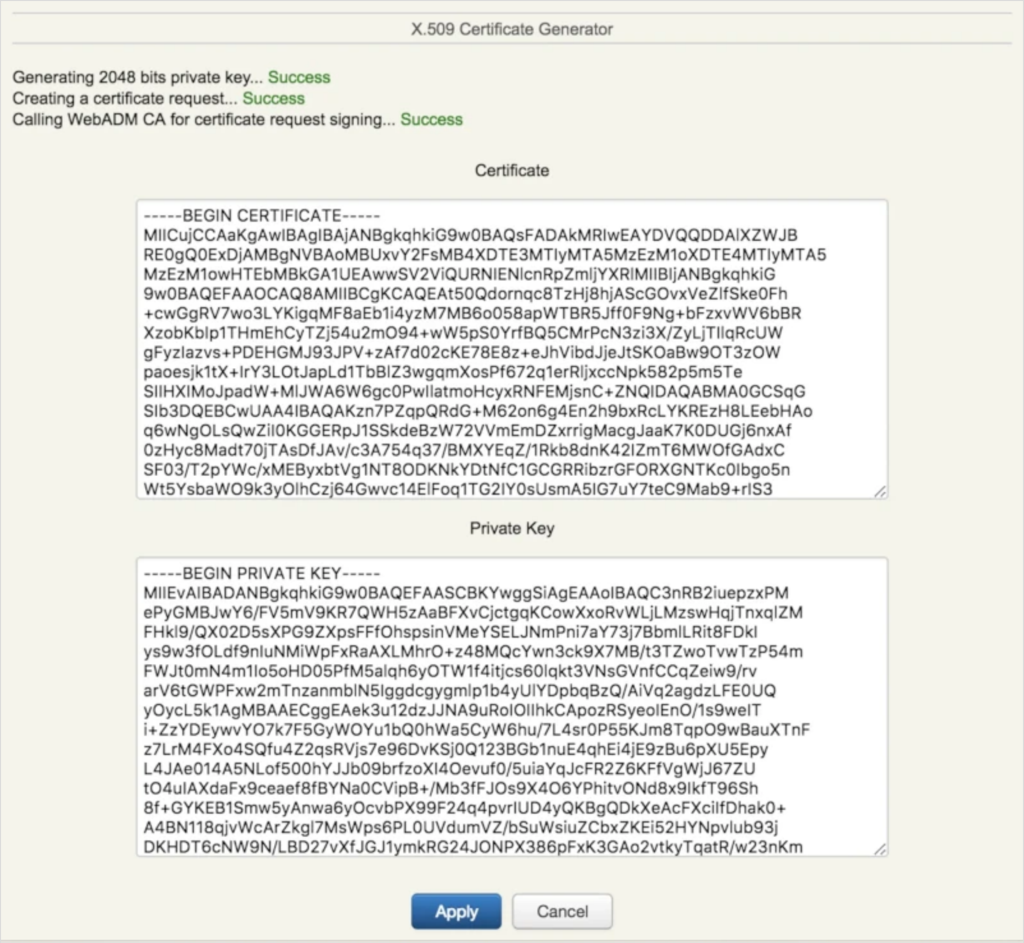


X.509 Certificate Generator

Common Name: WebADM Certificate

RSA Key Size: 2048 Bits

Now, we have the IdP certificate, we click on `Apply` and the `Server Certificate` and `Private key` will be auto filled in the configuration. You can also issue a certificate with your Entreprise CA if desired.



```
Generating 2048 bits private key... Success
Creating a certificate request... Success
Calling WebADM CA for certificate request signing... Success
```

Certificate

-----BEGIN CERTIFICATE-----
MIICujCCAaKgAwIBAgIBAjANBgkqhkiG9w0BAQsFADAkMRIwEAYDVQQDDAlXZWJB
RE0gQ0ExDjAMBgNVBAoMBUxvY2FsMB4XDTE3MTIyMTA5MzEzM1oXDTE4MTIyMTA5
MzEzM1owHTEbMBkGA1UEAwwSV2ViQURNIENlcnRpZmljYXRlMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt50Qdornqc8TzHj8hjAScGOvxVeZlfSke0Fh
+cwGgRV7wo3LYKigqMF8aEb1i4yzM7MB6o058apWTBR5Jff0F9Ng+bFzxvWV6bBR
XzobKbIp1THmEhCyTZj54u2mO94+wW5pS0YrfBQ5CMrPcN3zi3X/ZyLjTIlqRcUW
gFyzlazvs+PDEHGMJ93JPV+zAf7d02cKE78E8z+eJhVibdJjeJtSKOaBw9OT3zOW
paoesjk1tX+IrY3LOtJapLd1TbBlZ3wgqmXosPf672q1erRljxccNpk582p5m5Te
SIlHXIMoJpadW+MlJWA6W6gc0PwllatmoHcyxRNFEMjsnC+ZNQIDAQABMA0GCSqG
SIb3DQEBCwUAA4IBAQAKzn7PZqpQRdG+M62on6g4En2h9bxRcLYKREzH8LEebHAo
q6wNgOLsQwZil0KGGERpJ1SSkdeBzW72VVmEmDZxrrigMacgJaaK7K0DUGj6nxAf
0zHyc8Madt70jTAsDfJAv/c3A754q37/BMXYEqZ/1Rkb8dnK42lZmT6MWOfGAdxC
SF03/T2pYWc/xMEByxbtVg1NT8ODKNkYDtNfC1GCGRRibzrGFORXGNTKc0lbgo5n
Wt5YsbaWO9k3yOlhCzj64Gwvc14ElFoq1TG2lY0sUsmA5lG7uY7teC9Mab9+rIS3

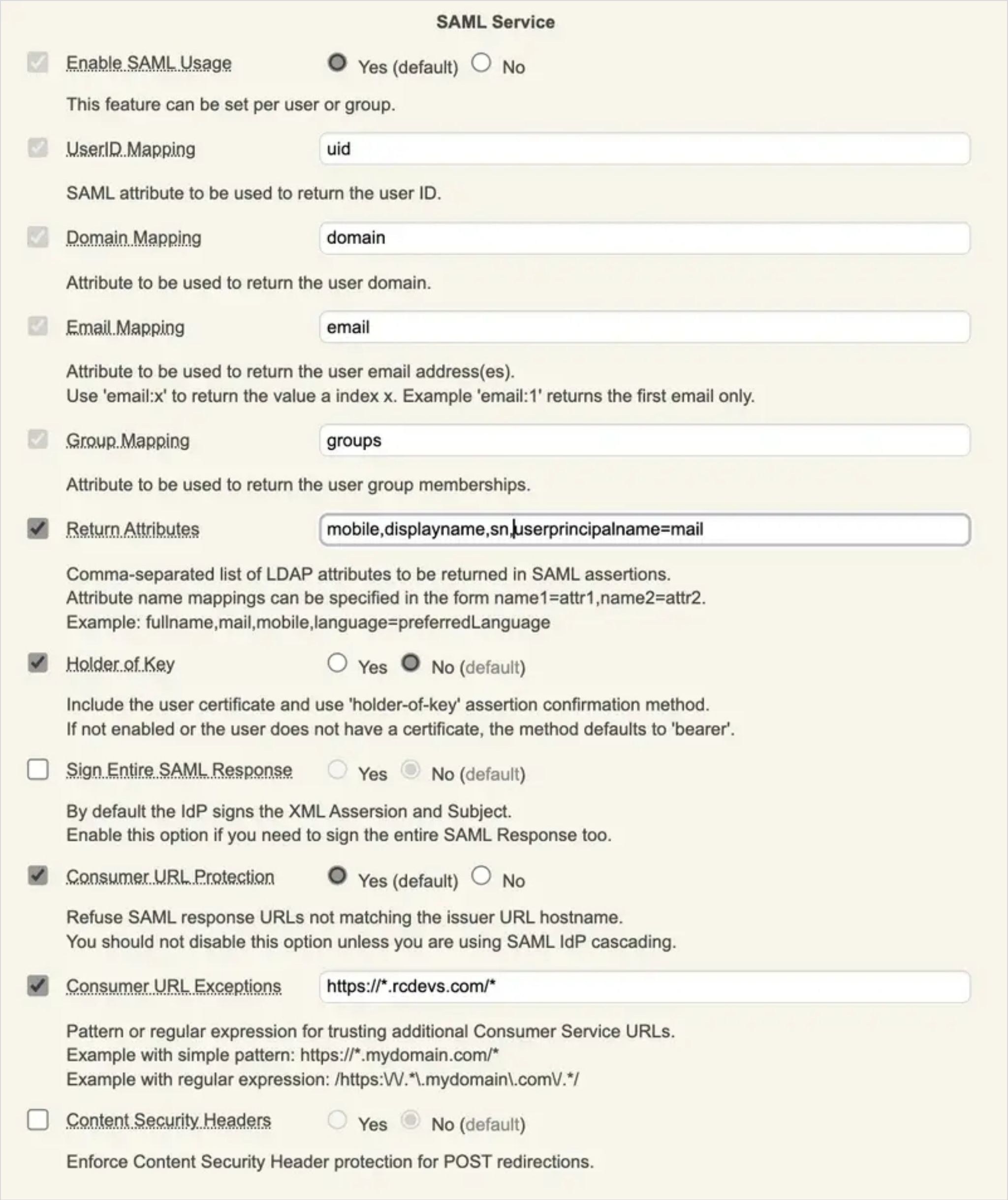
Private Key

-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC3nRB2iuepzxPM
ePyGMBJwY6/FV5mV9KR7QWH5zAaBFXvCjctgqKCowXxoRvWLjLMzswHqjTnxqlZM
FHkl9/QX02D5sXPG9ZXpsFFfOhspsinVMeYSELJNmPni7aY73j7BbmlLRit8FDkl
ys9w3fOLdf9nIuNMiWpFxRaAXLMhrO+z48MQcYwn3ck9X7MB/t3TZwoTvwTzP54m
FWJt0mN4m1Io5oHD05PfM5alqh6yOTW1f4itjcs60lqkt3VNsGVnfCCqZeiw9/rv
arV6tGWPFxw2mTnzanmblN5lggdcgygmlp1b4yUlYDpbqBzQ/AiVq2agdzLFE0UQ
yOycL5k1AgMBAAECggEAek3u12dzJJNA9uRolOIlhkCApozRSyeolEnO/1s9welT
i+ZzYDEywvYO7k7F5GyWOYu1bQ0hWa5CyW6hu/7L4sr0P55KJm8TqpO9wBauXTnF
z7LrM4FXo4SQfu4Z2qsRVjs7e96DvKSj0Q123BGb1nuE4qhEi4jE9zBu6pXU5Epy
L4JAe014A5NLof500hYJJb09brfzoXl4Oevuf0/5uiaYqJcFR2Z6KFfVgWjJ67ZU
tO4uIAXdaFx9ceaef8fBYNa0CVipB+/Mb3fFJOs9X4O6YPhitvONd8x9IkfT96Sh
8f+GYKEB1Smw5yAnwa6yOcvbPX99F24q4pvrIUD4yQKBgQDkXeAcFXcilfDhak0+
A4BN118qjvWcArZkgl7MsWps6PL0UVdumVZ/bSuWsiuZCbxZKEi52HYNpvlub93j
DKHDT6cNW9N/LBD27vXfJGJ1ymkRG24JONPX386pFxK3GAo2vtkyTqatR/w23nKm

The `Common Features` section is now configured.

2.2 SAML Configuration

We are now entering in the SAML dedicated configuration.



SAML Service

☑ Enable SAML Usage ● Yes (default) ○ No

This feature can be set per user or group.

☑ UserID Mapping uid

SAML attribute to be used to return the user ID.

☑ Domain Mapping domain

Attribute to be used to return the user domain.

☑ Email Mapping email

Attribute to be used to return the user email address(es).
Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

☑ Group Mapping groups

Attribute to be used to return the user group memberships.

☑ Return Attributes mobile,displayname,sn,userprincipalname=mail

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

☑ Holder of Key ○ Yes ● No (default)

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

□ Sign Entire SAML Response ○ Yes ● No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

☑ Consumer URL Protection ● Yes (default) ○ No

Refuse SAML response URLs not matching the issuer URL hostname.
You should not disable this option unless you are using SAML IdP cascading.

☑ Consumer URL Exceptions https://*.rcdevs.com/*

Pattern or regular expression for trusting additional Consumer Service URLs.
Example with simple pattern: https://*.mydomain.com/*
Example with regular expression: /https:VV.*\.mydomain\.comV.*/

□ Content Security Headers ○ Yes ● No (default)

Enforce Content Security Header protection for POST redirections.

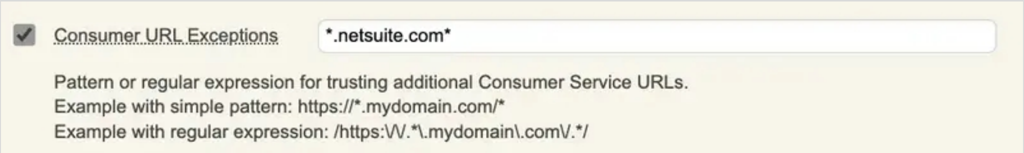
- The `Enable SAML Usage` setting enable the SAML configuration in order to implement SP through SAML.
- The `UserID Mapping` setting is the attribut value used in the SAML response to return the user ID.
- The `Domain Mapping` setting is the attribut value used in the SAML response to return the domain value. By default, the WebADM domain name is returned based on the domain used to authenticate the user.
- The `Email Mapping` setting is the attribut value used in the SAML response to return the users' email value(s).
- The `Group Mapping` setting is the attribut value used in the SAML response to return the user group memberships.
- The `Return attributes` setting is the attribut value used in the SAML response to return a list of desired attributs. You can also manipulate values returned. For example here, I returned in SAML response mobile, displayname sn attributs retrieved from the LDAP account and in userprincipalname I put the user email value.
- The `Holder of Key` setting is used to include the user certificate and use 'holder-of-key' assertion confirmation method. If not enabled or the user does not have a certificate, the method defaults to 'bearer'.
- The `Sign Entire SAML Response` setting is used to intirely sign the SAML response. This can be an option on some service provider. By default, the IdP signs the XML assersion and the subject.

- The `Consumer URL Protection` is a security setting used to refuse SAML requests containing AssertionConsumerServiceURL which do not match the `Issuer URL` hostname present in the same request.
- The `Consumer URL Exception` setting can be used when the AssertionConsumerServiceURL present in the SAML request do not match the SP issuer URL.

example:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:AuthnRequest AssertionConsumerServiceURL="https://system.netsuite.com/saml2/acs"
                     Destination="https://waproxy.support.rcdevs.com/openid/index.php"
                     ForceAuthn="false"
                     ID="_184481c4dc4698ff64574278aa43d60"
                     IsPassive="false"
                     IssueInstant="2023-11-09T14:26:25.059Z"
                     ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                     Version="2.0"
                     xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml2:Issuer
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">http://www.netsuite.com/sp</saml2:Issuer>
  <saml2p:NameIDPolicy AllowCreate="true"
                       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
                       SPNameQualifier="http://www.netsuite.com/sp" /></saml2p:AuthnRequest>
```

In that example, the AssertionConsumerService URL hostame (system.netsuite.com) do not match the Issuer hostname (netsuite.com). I can then configure a `Consumer URL Exceptions` like this:



☑ Consumer URL Exceptions: *.netsuite.com*

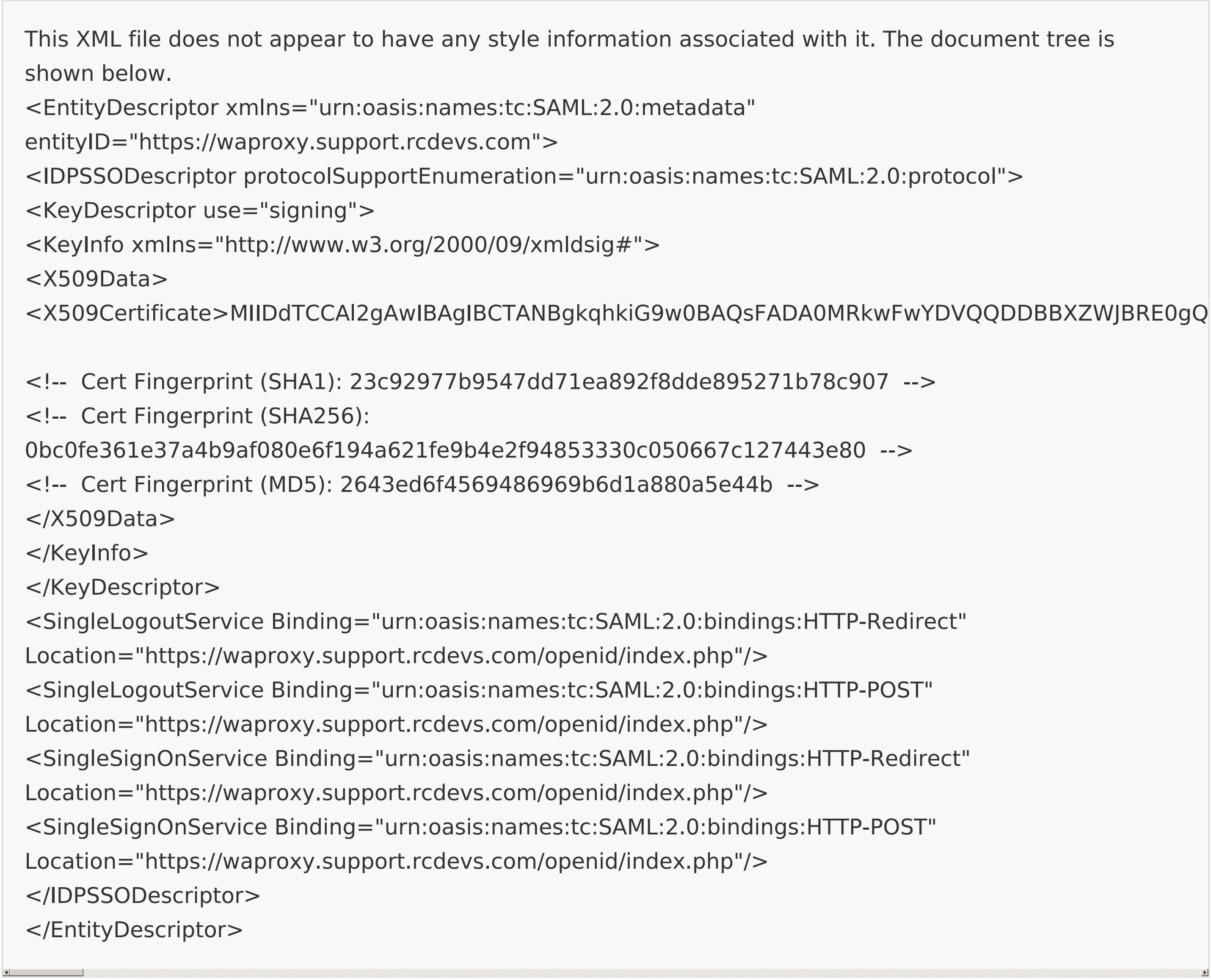
Pattern or regular expression for trusting additional Consumer Service URLs.
Example with simple pattern: https://*.mydomain.com/*
Example with regular expression: /https:\/\/.*\.mydomain\.com\/.*/

By default, the AssertionConsumerServiceURL is taken from the SAML request and is used by ther IDP after the user authentication to send the response to the service provider. The AssertionConsumerServiceURL can be rewrite by client policies if needed. If multiple AssertionConsumerServiceURL are available on your service provider, then you can also use the `Consumer URL Exception` and configure a regex that will match all URLs.

- The `Content Security Headers` setting can be used to enforce content security header protection for POST redirections.

You can now save your SAML configuration. The SAML metadata URL is accessible through WebADM servers and through WAProxy servers if the Web Application is published through WAPRoxy:

- Metadata URL from the WebADM server: https://webadm1.support.rcdevs.com/webapps/openid/metadata/
- Metadata URL from the WAProxy: https://waproxy.support.rcdevs.com/ws/saml/



```
This XML file does not appear to have any style information associated with it. The document tree is
shown below.
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://waproxy.support.rcdevs.com">
<IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIDdTCCAl2gAwIBAgIBCTANBgkqhkiG9w0BAQsFADA0MRkwFwYDVQQDDBBXZWJBRE0gQ

<!--  Cert Fingerprint (SHA1): 23c92977b9547dd71ea892f8dde895271b78c907  -->
<!--  Cert Fingerprint (SHA256):
0bc0fe361e37a4b9af080e6f194a621fe9b4e2f94853330c050667c127443e80  -->
<!--  Cert Fingerprint (MD5): 2643ed6f4569486969b6d1a880a5e44b  -->
</X509Data>
</KeyInfo>
</KeyDescriptor>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
</IDPSSODescriptor>
</EntityDescriptor>
```

The SAML clients (Service Providers) need to know about the SAML IdP endpoints. Most clients will accept the autoconfiguration with an XML-based metadata URL. You can provide the previous URLs according to your scenario.

⚠ Important

Many SAML Service Providers will require your WebADM to be run with a trusted SSL certificate. To use your own trusted certificate and key, please have a look on Trusted Certificate documentation.

2.3 OpenID Configuration

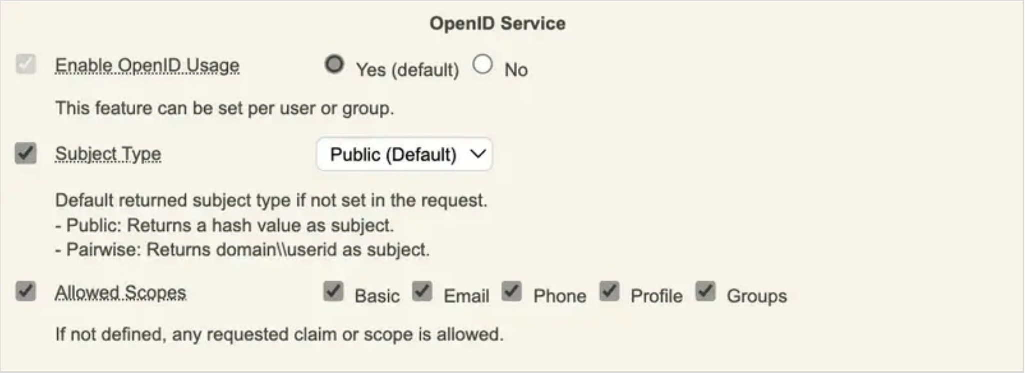
The configuration of OpenID service is very simple. Version 1.2x includes the support for OpenID-Connect and OAuth2.

To use your identity provider in OpenID-Connect mode, the client configuration must pass the scope ‘openid’ in the IdP requests. The supported OpenID-Connect scopes are: basic, email, phone, profile and groups.

To use your identity provider in OAuth2 mode, the client must pass the scope ‘profile’ in the IdP requests.

If your client application needs the user's email address, you can additionally request the openid email scope.

The Allwed scopes must be enabled in the global configuration or per client policy in order to be returned to the service providers which are requesting them in their request.



OpenID Service

☑ Enable OpenID Usage ● Yes (default) ○ No

This feature can be set per user or group.

☑ Subject Type Public (Default)

Default returned subject type if not set in the request.
- Public: Returns a hash value as subject.
- Pairwise: Returns domain\\userid as subject.

☑ Allowed Scopes ☑ Basic ☑ Email ☑ Phone ☑ Profile ☑ Groups

If not defined, any requested claim or scope is allowed.

The OpenID metadata URL is accessible through WebADM servers and through WAProxy servers if the Web Application is published through WAPRoxy:

- Metadata URL from the WebADM server: https://webadm1.support.rcdevs.com/webapps/openid/.well-known/openid-configuration
- Metadata URL from the WAProxy: https://waproxy.support.rcdevs.com/ws/openid/

Which is returning the following in my scenario:

```
{
    "issuer": "https://waproxy.support.rcdevs.com",
    "authorization_endpoint": "https://waproxy.support.rcdevs.com/openid/index.php",
    "token_endpoint": "https://waproxy.support.rcdevs.com/openid/index.php",
    "userinfo_endpoint": "https://waproxy.support.rcdevs.com/openid/index.php",
    "jwks_uri": "https://waproxy.support.rcdevs.com/openid/certs.php",
    "subject_types_supported": [
        "public",
        "pairwise"
    ],
    "response_types_supported": [
        "code",
        "token",
        "id_token"
    ],
    "response_modes_supported": [
        "query",
        "fragment",
        "form_post"
    ],
    "id_token_signing_alg_values_supported": [
        "RS256"
    ],
    "scope_supported": [
        "basic",
        "openid",
        "email",
        "phone",
        "profile",
        "groups"
    ],
    "claims_supported": [
        "sub",
        "email",
        "email_verified",
        "phone_number",
        "phone_number_verified",
        "preferred_username",
        "preferred_language",
        "given_name",
        "family_name",
        "name",
        "groups",
        "mfa-policy"
    ]
}
```

3. Configuration of a Service Provider

3.1 IDP initiated (SAML)

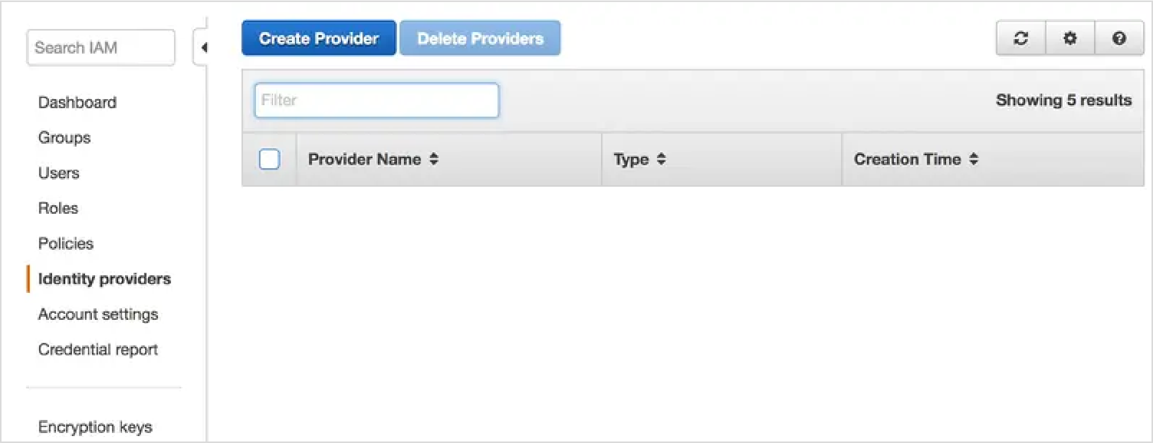
In this scenario, the authentication will be started directly from *OpenID & SAML Provider* web application. We will configure WebADM to manage authentications with Amazon Web Service (AWS). Other Service providers are available but not shown in this HowTo: GSuite, SalesForce, SugarCRM, Zimbra, GoToMeeting, GoToWebinar, GoToTraining and GoToAssist.

3.1.1 AWS SAML integration

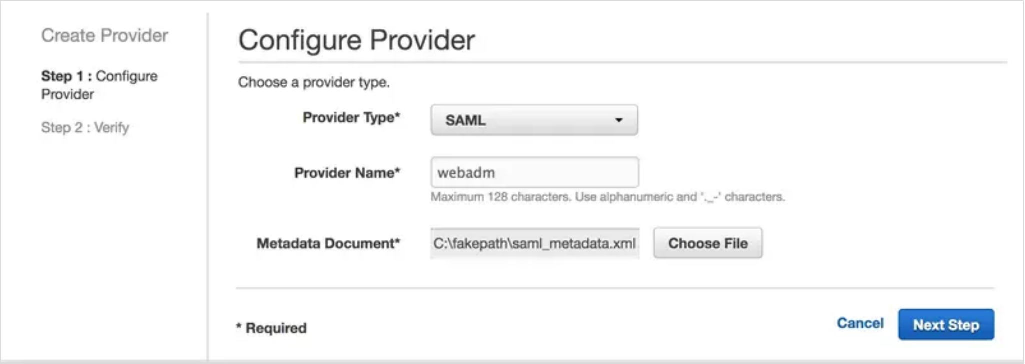
3.1.1.1 SAML Configuration on AWS

First, we save the SAML metadata in a file. For our IdP server, we find it in `https://webadm.local/ws/saml/`.

We open AWS console > `IAM` > `Identity providers` > _`Create Provider`:



We select `SAML`, add a name, insert the metadata file and click on `Next Step`:



Provider Type*: SAML

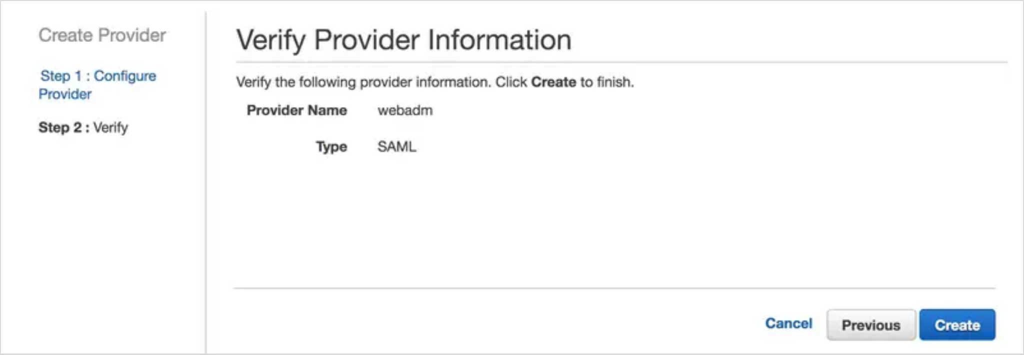
Provider Name*: webadm

Maximum 128 characters. Use alphanumeric and '._-' characters.

Metadata Document*: C:\fakepath\saml_metadata.xml Choose File

* Required

We click on `Create`:



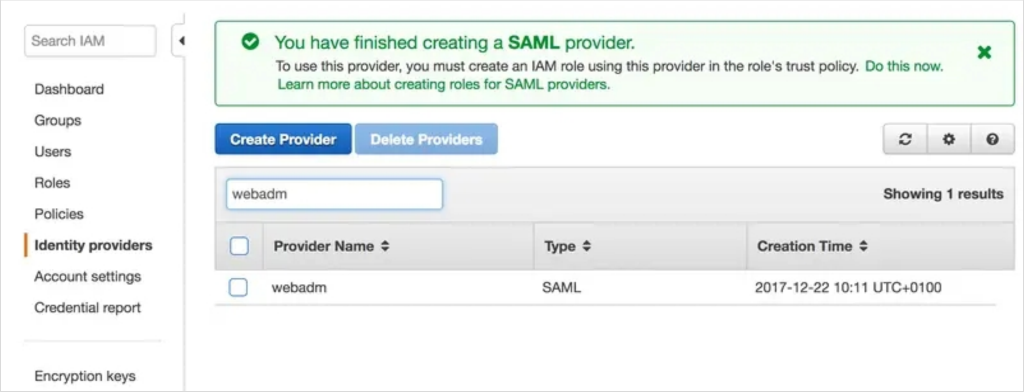
Verify Provider Information

Verify the following provider information. Click **Create** to finish.

Provider Name webadm

Type SAML

Now, our IdP is added to AWS. We select `Roles`:

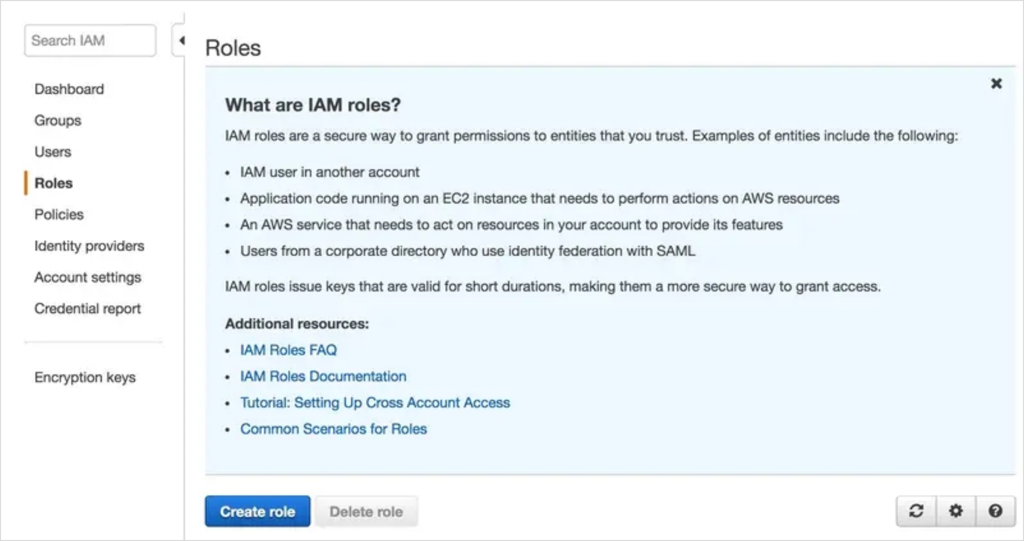


You have finished creating a **SAML** provider.
To use this provider, you must create an IAM role using this provider in the role's trust policy. Do this now.
Learn more about creating roles for SAML providers.

webadm — **Showing 1 results**

□	Provider Name	Type	Creation Time
□	webadm	SAML	2017-12-22 10:11 UTC+0100

We click on `Create Role`:



Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

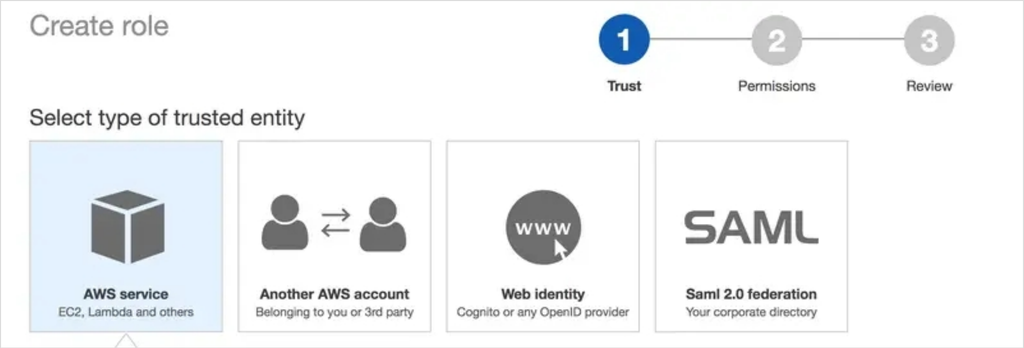
- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

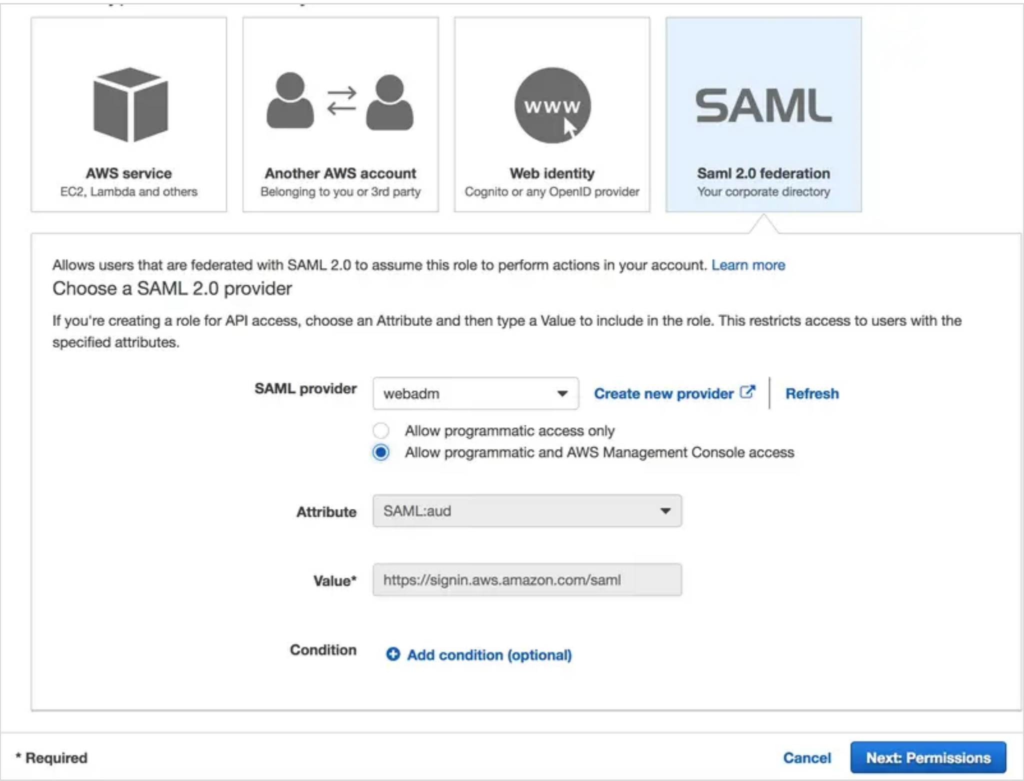
We click on `SAML`:



Select type of trusted entity

- AWS service — EC2, Lambda and others
- Another AWS account — Belonging to you or 3rd party
- Web identity — Cognito or any OpenID provider
- Saml 2.0 federation — Your corporate directory

We select our SAML provider, select `AWS Management Console access` and click on `Next Permission`:



- AWS service — EC2, Lambda and others
- Another AWS account — Belonging to you or 3rd party
- Web identity — Cognito or any OpenID provider
- Saml 2.0 federation — Your corporate directory

Allows users that are federated with SAML 2.0 to assume this role to perform actions in your account. Learn more

Choose a SAML 2.0 provider

If you're creating a role for API access, choose an Attribute and then type a Value to include in the role. This restricts access to users with the specified attributes.

SAML provider: webadm

○ Allow programmatic access only

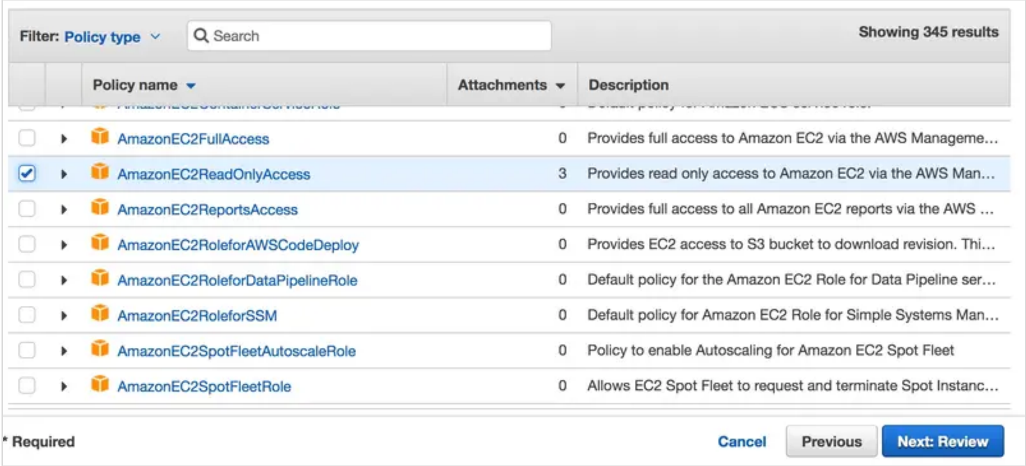
● Allow programmatic and AWS Management Console access

Attribute: SAML:aud

Value*: https://signin.aws.amazon.com/saml

Condition

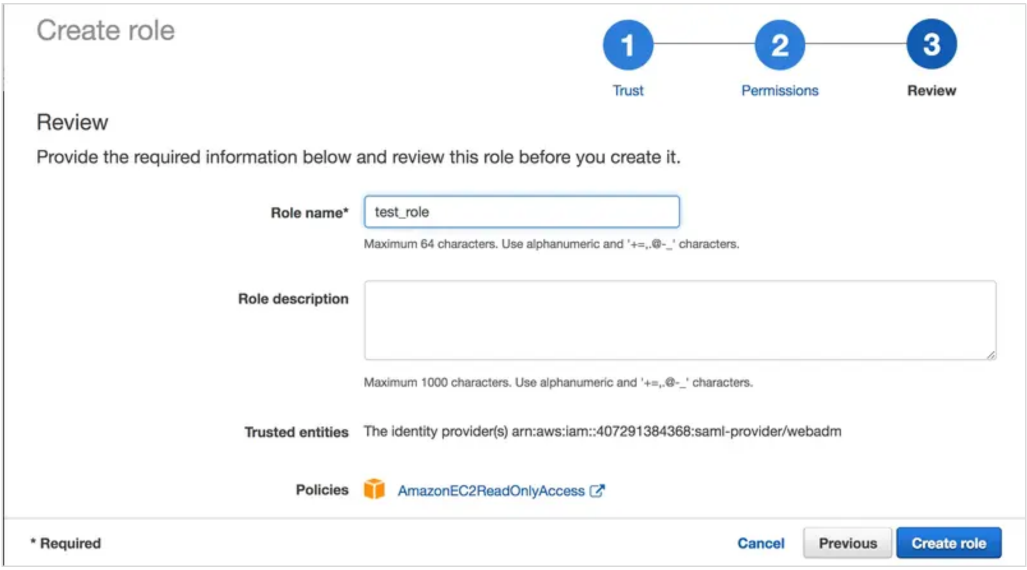
We select a permission policy and click on `Next: Review`.



Filter: Policy type | Search | Showing 345 results

		Policy name	Attachments	Description
□	▸	AmazonEC2FullAccess	0	Provides full access to Amazon EC2 via the AWS Manageme...
☑	▸	AmazonEC2ReadOnlyAccess	3	Provides read only access to Amazon EC2 via the AWS Man...
□	▸	AmazonEC2ReportsAccess	0	Provides full access to all Amazon EC2 reports via the AWS ...
□	▸	AmazonEC2RoleforAWSCodeDeploy	0	Provides EC2 access to S3 bucket to download revision. Thi...
□	▸	AmazonEC2RoleforDataPipelineRole	0	Default policy for the Amazon EC2 Role for Data Pipeline ser...
□	▸	AmazonEC2RoleforSSM	0	Default policy for Amazon EC2 Role for Simple Systems Man...
□	▸	AmazonEC2SpotFleetAutoscaleRole	0	Policy to enable Autoscaling for Amazon EC2 Spot Fleet
□	▸	AmazonEC2SpotFleetRole	0	Allows EC2 Spot Fleet to request and terminate Spot Instanc...

We add a name and click on `Create role`:



Review

Provide the required information below and review this role before you create it.

Role name* test_role

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

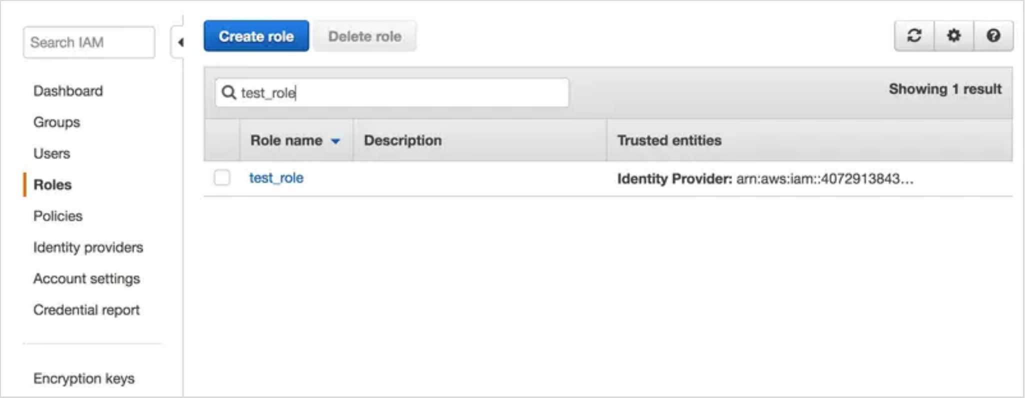
Role description

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities The identity provider(s) arn:aws:iam::407291384368:saml-provider/webadm

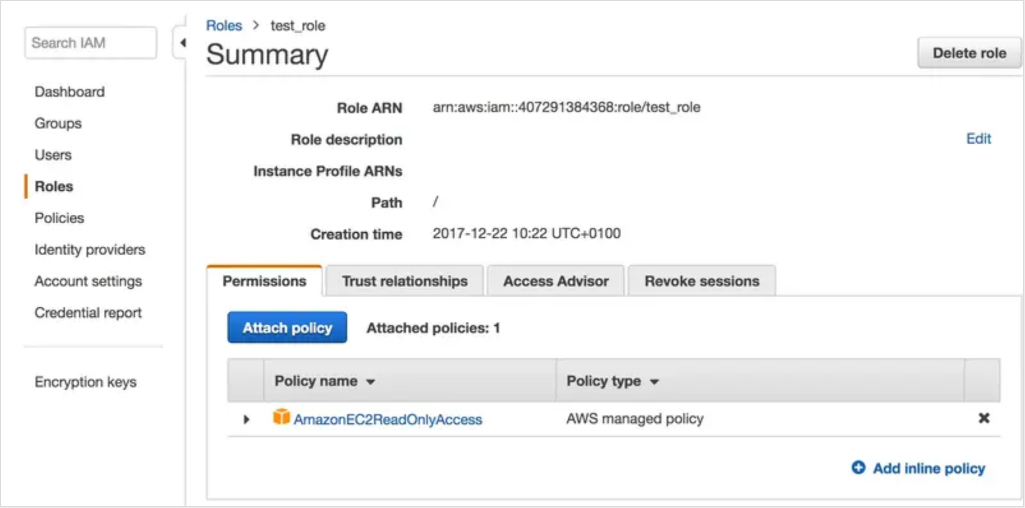
Policies AmazonEC2ReadOnlyAccess

The role is now created, we can select it to see more details.



test_role Showing 1 result

Role name	Description	Trusted entities
test_role		Identity Provider: arn:aws:iam::4072913843...



Role ARN arn:aws:iam::407291384368:role/test_role

Role description Edit

Instance Profile ARNs

Path /

Creation time 2017-12-22 10:22 UTC+0100

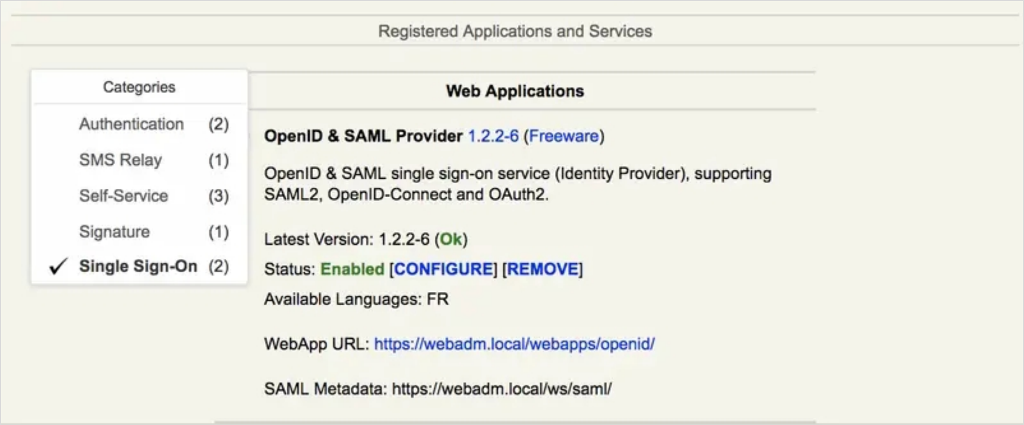
Attached policies: 1

Policy name	Policy type
AmazonEC2ReadOnlyAccess	AWS managed policy

3.1.1.2 Configure WebADM IDP for AWS

We need to activate IdP initiated authentication for AWS.

We open the configuration in WebADM GUI > `Applications` > `Single Sign-on` > `CONFIGURE`:



Registered Applications and Services

Categories	
Authentication	(2)
SMS Relay	(1)
Self-Service	(3)
Signature	(1)
✓ **Single Sign-On**	(2)

Web Applications

OpenID & SAML Provider 1.2.2-6 (Freeware)

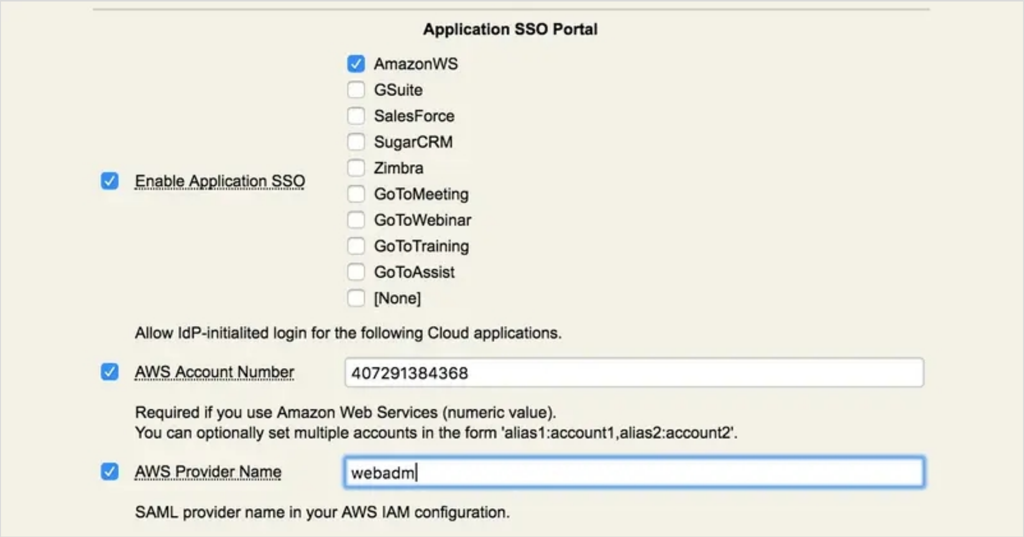
OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.2.2-6 (Ok)
Status: **Enabled** **[CONFIGURE]** **[REMOVE]**
Available Languages: FR

WebApp URL: https://webadm.local/webapps/openid/

SAML Metadata: https://webadm.local/ws/saml/

We check `Enable Application SSO` and `AmazonWS`, we add `AWS Account Number` (a numerical value that you can find in the ARN of the AWS role) and `AWS Provider Name` and apply:



Application SSO Portal

☑ Enable Application SSO

☑ AmazonWS
☐ GSuite
☐ SalesForce
☐ SugarCRM
☐ Zimbra
☐ GoToMeeting
☐ GoToWebinar
☐ GoToTraining
☐ GoToAssist
☐ [None]

Allow IdP-initialited login for the following Cloud applications.

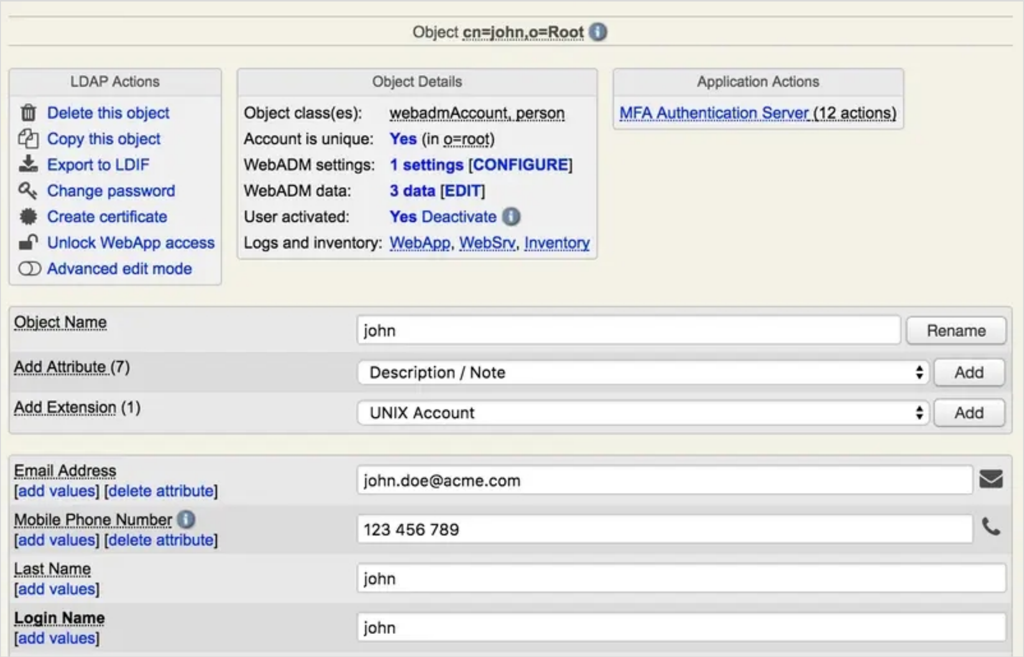
☑ AWS Account Number: 407291384368

Required if you use Amazon Web Services (numeric value).
You can optionally set multiple accounts in the form 'alias1:account1,alias2:account2'.

☑ AWS Provider Name: webadm

SAML provider name in your AWS IAM configuration.

We select the test user and click on `WebADM settings: [CONFIGURE]`:



Object cn=john,o=Root

LDAP Actions
- Delete this object
- Copy this object
- Export to LDIF
- Change password
- Create certificate
- Unlock WebApp access
- Advanced edit mode

Object Details

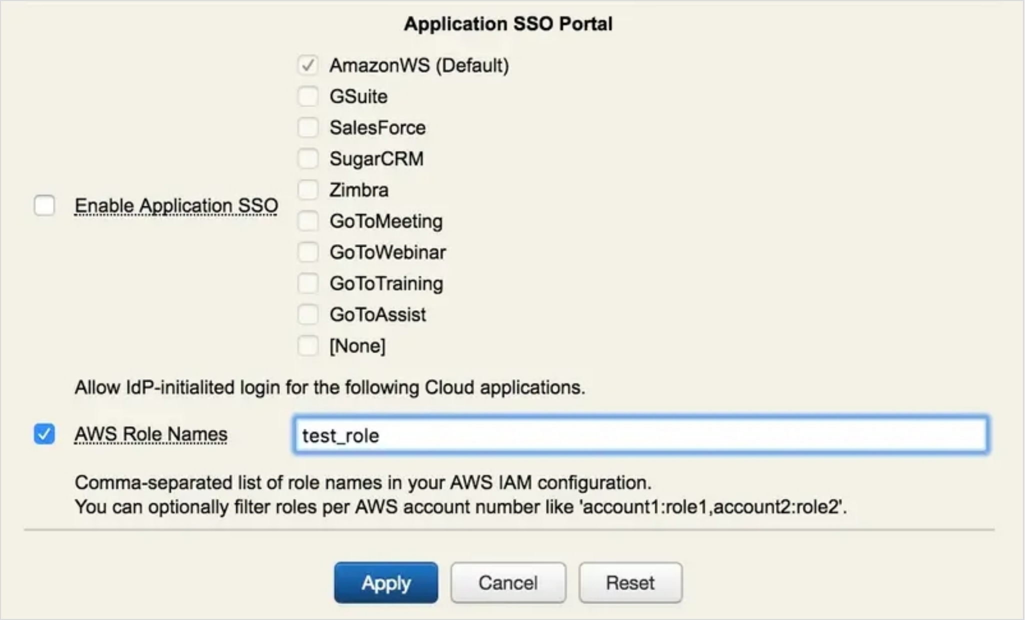
Object class(es):	webadmAccount, person
Account is unique:	**Yes** (in o=root)
WebADM settings:	**1 settings [CONFIGURE]**
WebADM data:	**3 data [EDIT]**
User activated:	**Yes** Deactivate
Logs and inventory:	WebApp, WebSrv, Inventory

Application Actions
- MFA Authentication Server (12 actions)

Object Name	john [Rename]
Add Attribute (7)	Description / Note [Add]
Add Extension (1)	UNIX Account [Add]

Email Address [add values] [delete attribute]	john.doe@acme.com
Mobile Phone Number [add values] [delete attribute]	123 456 789
Last Name [add values]	john
Login Name [add values]	john

We select `OpenID`, add `AWS Role Names` and `Apply`. We can also add the AWS role to an LDAP group:



Application SSO Portal

□ Enable Application SSO

☑ AmazonWS (Default)
□ GSuite
□ SalesForce
□ SugarCRM
□ Zimbra
□ GoToMeeting
□ GoToWebinar
□ GoToTraining
□ GoToAssist
□ [None]

Allow IdP-initialited login for the following Cloud applications.

☑ AWS Role Names: test_role

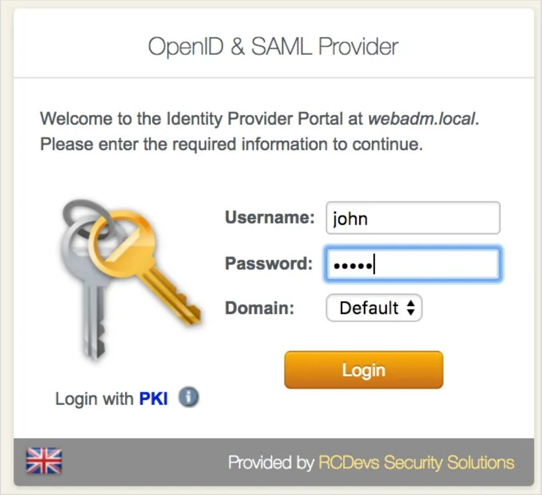
Comma-separated list of role names in your AWS IAM configuration.
You can optionally filter roles per AWS account number like 'account1:role1,account2:role2'.

3.1.1.3 AWS users/groups/clients policies

See more in section `4. How to create and match a client policy per service provider`. The example used is with AWS.

3.1.1.4 Testing/Debug

To test, open the web application in `https://webadm.local/webapps/openid/` and `Login` with the user:

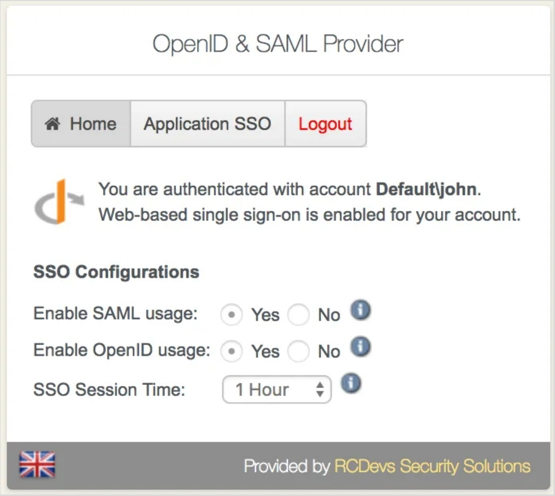


OpenID & SAML Provider

Welcome to the Identity Provider Portal at *webadm.local*.
Please enter the required information to continue.

Username: john
Password: •••••
Domain: Default

We select `Application SSO`:



You are authenticated with account **Default\john**.
Web-based single sign-on is enabled for your account.

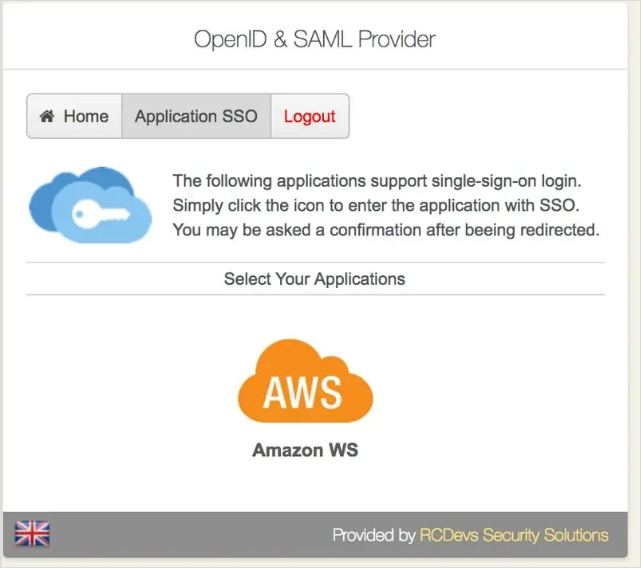
SSO Configurations

Enable SAML usage: ● Yes ○ No

Enable OpenID usage: ● Yes ○ No

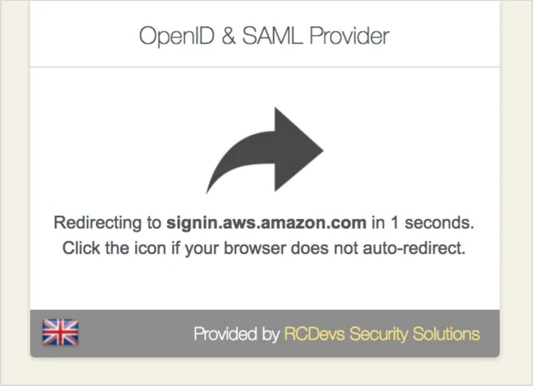
SSO Session Time: 1 Hour

We click on `Amazon WS`:



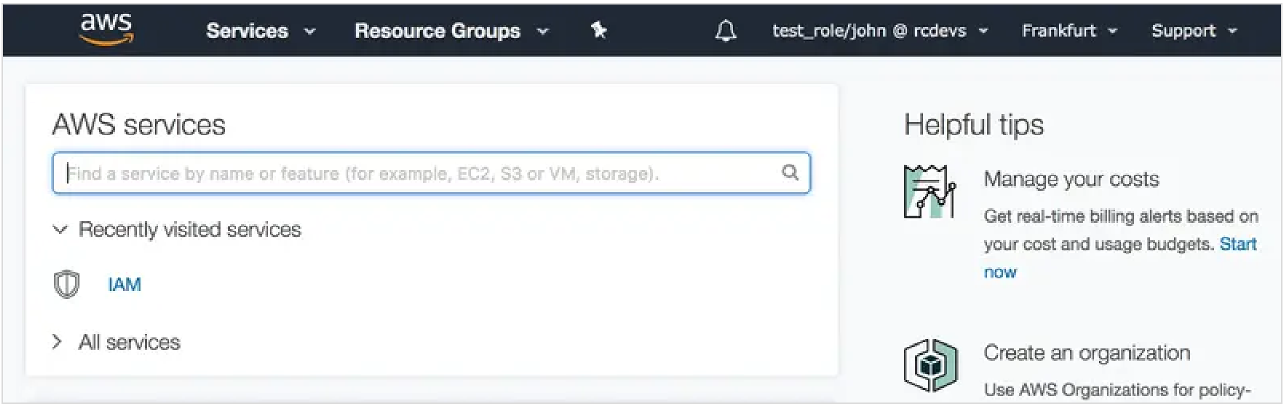
The following applications support single-sign-on login.
Simply click the icon to enter the application with SSO.
You may be asked a confirmation after beeing redirected.

Select Your Applications



Redirecting to **signin.aws.amazon.com** in 1 seconds.
Click the icon if your browser does not auto-redirect.

That’s it,we are now connected to AWS:



Get real-time billing alerts based on your cost and usage budgets. Start now

Use AWS Organizations for policy-

We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] New login request (OpenOTP)
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Username: john
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Domain: Default
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > ANY Password: xxxxxxx
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] Sending openotpSimpleLogin request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] New openotpSimpleLogin SOAP request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Username: john
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Domain: Default
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Password: xxxxxxx
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Client ID: OpenID
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Source IP: 192.168.1.220
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Registered openotpSimpleLogin request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Resolved LDAP user: cn=john,o=Root (cached)
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Started transaction lock for user
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user mobiles: 123 456 789
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user emails: john.doe@acme.com
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTimeou
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=

[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 2 user data: LoginCount,RejectCount
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Requested login factors: LDAP
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] LDAP password Ok
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Updated user data
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Sent success response
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] OpenOTP authentication success
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Resolved LDAP user: cn=john,o=Root
(cached)
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Login session started for cn=john,o=Root
[2017-12-22 09:36:50] [192.168.1.220] [OpenID:4JGOGC0T] Sent SAML success response
```

3.2 SP-Initiated (SAML)

3.2.1 SimpleSAMLPHP

For this test, we are using simplesamplphp.

We install it on another *CentOS 7* server.

We open http port:

```
firewall-cmd --permanent --add-service http
firewall-cmd --reload
```

We disable selinux:

```
setenforce 0
vi /etc/selinux/config
```

We install required packages:

```
yum install wget php php-mbstring php-xml httpd
```

We install *simplesamlphp*:

```
wget "https://simplesamlphp.org/download?latest" -O ssp.tgz
tar xzf ssp.tgz
mv simplesamlphp* /var/simplesamlphp
```

We add a virtual host to *Apache* (replace *sp.local* with the right DNS name who point to this server):

```
vi /etc/httpd/conf.d/saml.conf
```

```
<VirtualHost *>
     ServerName sp.local
     DocumentRoot /var/www/sp.local

     SetEnv SIMPLESAMLPHP_CONFIG_DIR /var/simplesamlphp/config

     Alias /simplesaml /var/simplesamlphp/www

     <Directory /var/simplesamlphp/www>
        Require all granted
     </Directory>
</VirtualHost>
```

We add the Identity Provider. All these values should correspond to the content of metadata from SAML configuration in WebADM:

- *$metadata* corresponds to *entityID*
- *SingleSignOnService* corresponds to *SingleSignOnService Location=*
- *SingleLogoutService* corresponds to *SingleLogoutService Location=*
- *certFingerprint* corresponds to *Cert Fingerprint (SHA1)*

```
vi /var/simplesamlphp/metadata/saml20-IdP-remote.php
```

```
<?php
 $metadata['https://webadm.local'] = array(
  'SingleSignOnService'  => 'https://webadm.local/webapps/openid/',
  'SingleLogoutService'  => 'https://webadm.local/webapps/openid/',
  'certFingerprint'      => '802b0a629dfc11a686306a73f8b11b272e1b9ca2',
);
```

We enable *SAML* in `/var/simplesamlphp/config/config.php`:

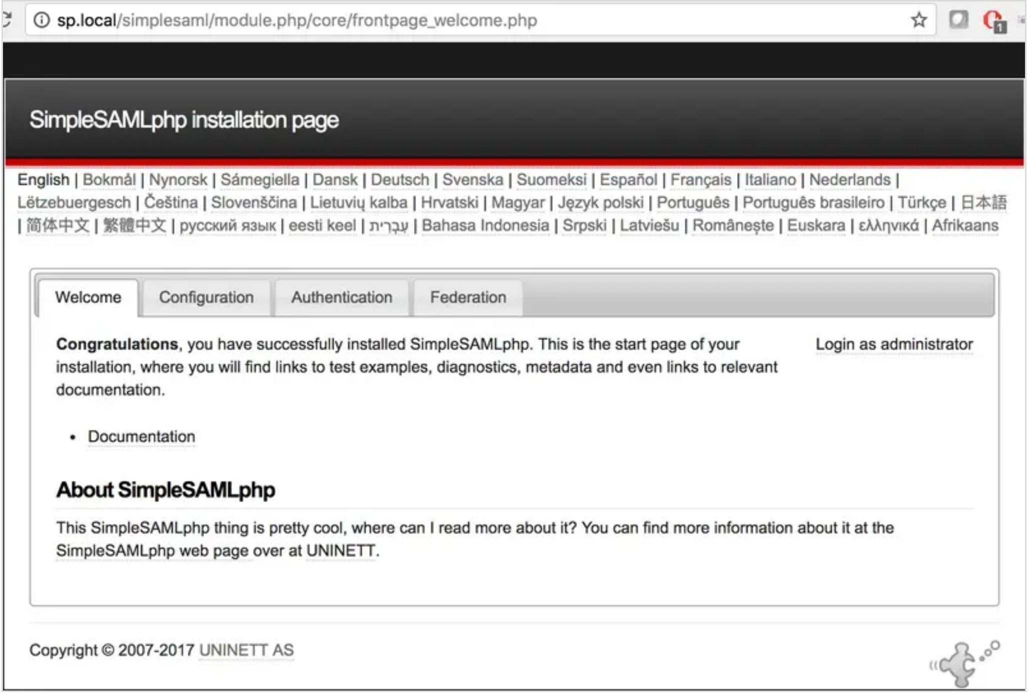
```
vi /var/simplesamlphp/config/config.php
```

```
enable.saml20-IdP' => true
```

We start *Apache*:

```
systemctl start httpd
systemctl enable httpd
```

We open `http://sp.local/simplesaml` in a browser:



sp.local/simplesaml/module.php/core/frontpage_welcome.php

SimpleSAMLphp installation page

English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עִבְרִית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara | ελληνικά | Afrikaans

Congratulations, you have successfully installed SimpleSAMLphp. This is the start page of your installation, where you will find links to test examples, diagnostics, metadata and even links to relevant documentation.

Login as administrator

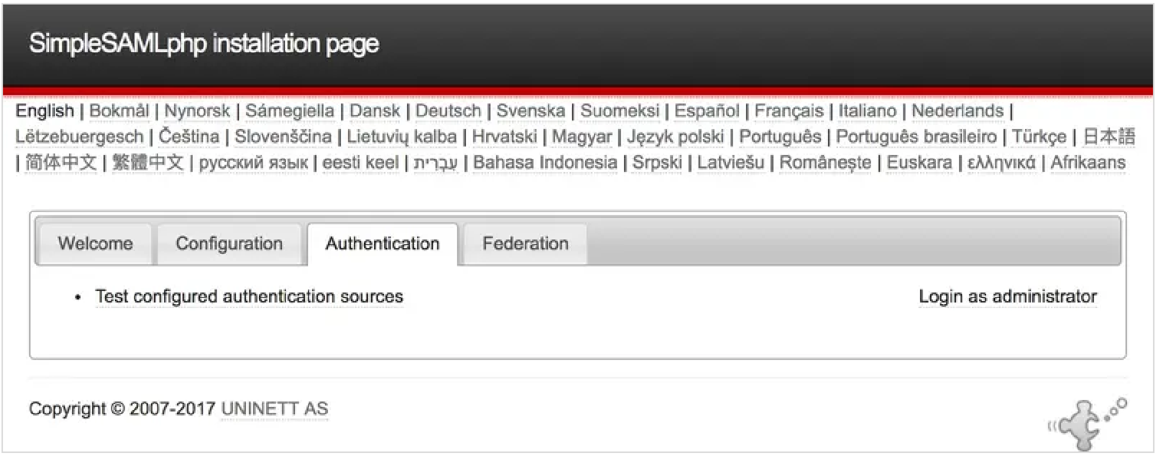
- Documentation

About SimpleSAMLphp

This SimpleSAMLphp thing is pretty cool, where can I read more about it? You can find more information about it at the SimpleSAMLphp web page over at UNINETT.

Copyright © 2007-2017 UNINETT AS

We click on `Authentication`:



SimpleSAMLphp installation page

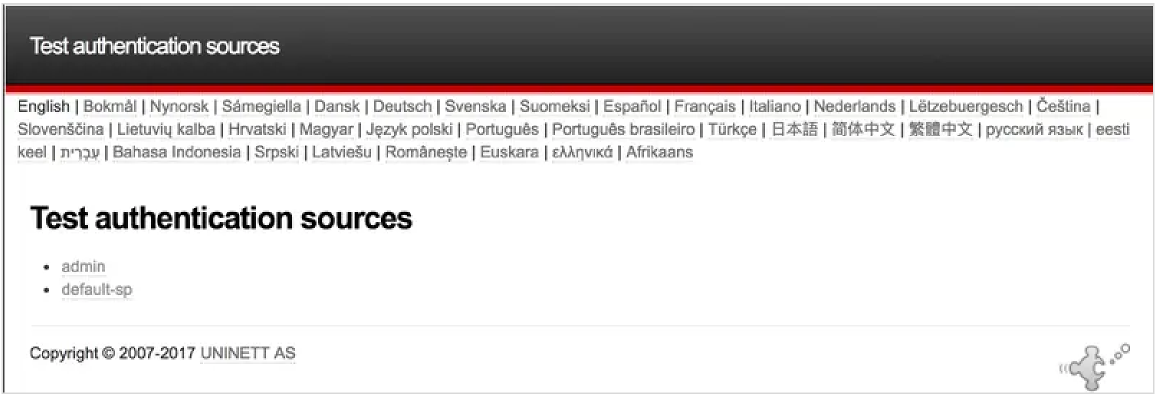
English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עִבְרִית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara | ελληνικά | Afrikaans

- Test configured authentication sources

Login as administrator

Copyright © 2007-2017 UNINETT AS

We click on `Test configured authentication sources`:



Test authentication sources

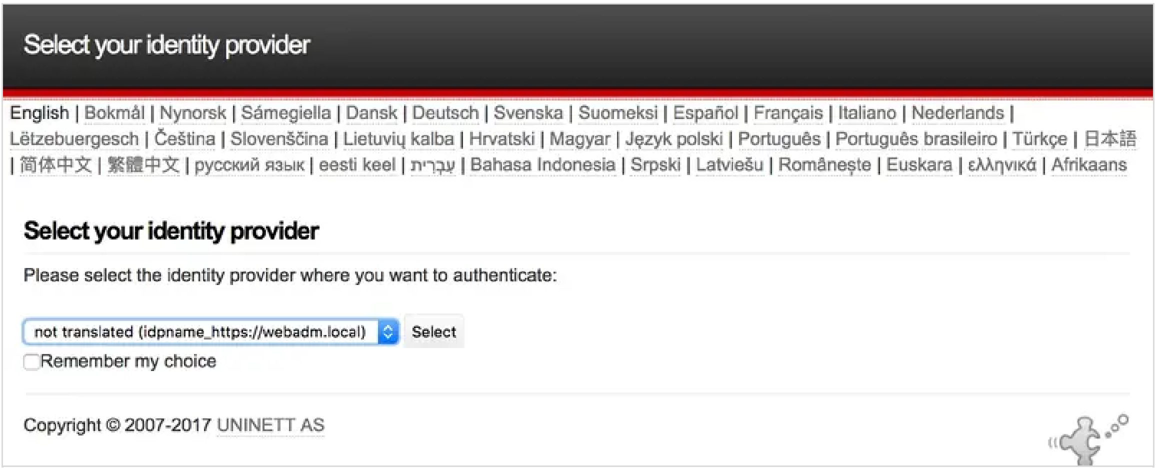
English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עִבְרִית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara | ελληνικά | Afrikaans

Test authentication sources

- admin
- default-sp

Copyright © 2007-2017 UNINETT AS

We click on `default-sp`:



Select your identity provider

English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עִבְרִית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara | ελληνικά | Afrikaans

Select your identity provider

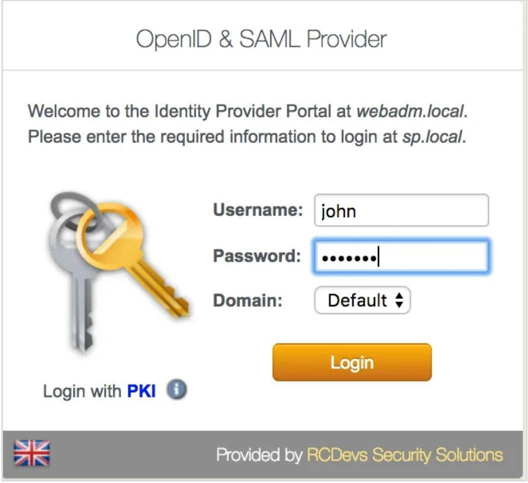
Please select the identity provider where you want to authenticate:

not translated (idpname_https://webadm.local) Select

□ Remember my choice

Copyright © 2007-2017 UNINETT AS

We click on `Select`:



OpenID & SAML Provider

Welcome to the Identity Provider Portal at *webadm.local*.
Please enter the required information to login at *sp.local*.

Username: john

Password: •••••••

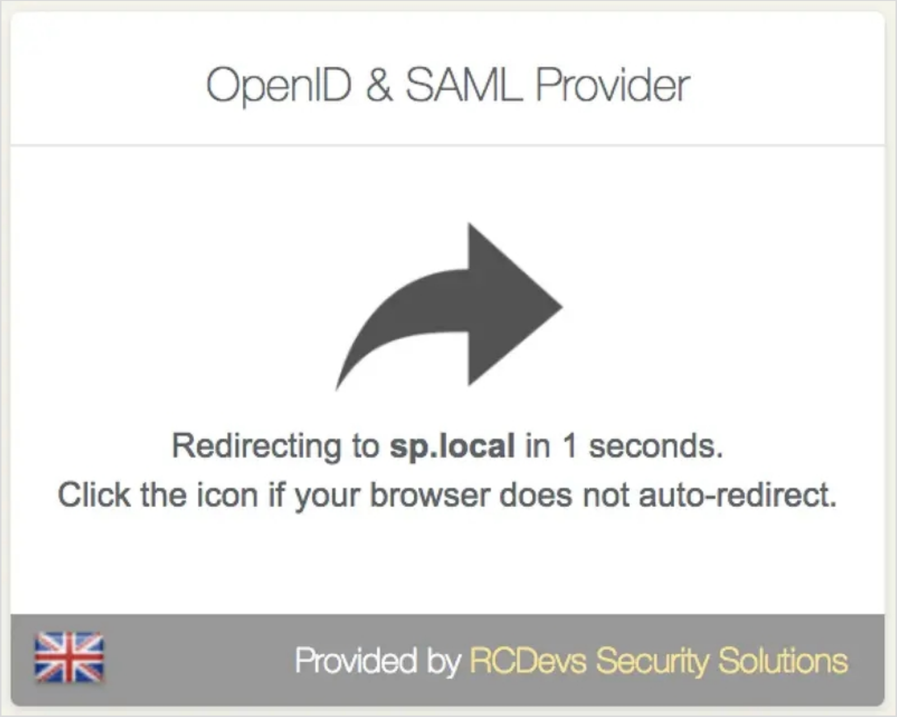
Domain: Default

Login

Login with PKI

Provided by RCDevs Security Solutions

We authenticate with an activated user through WebADM IdP:

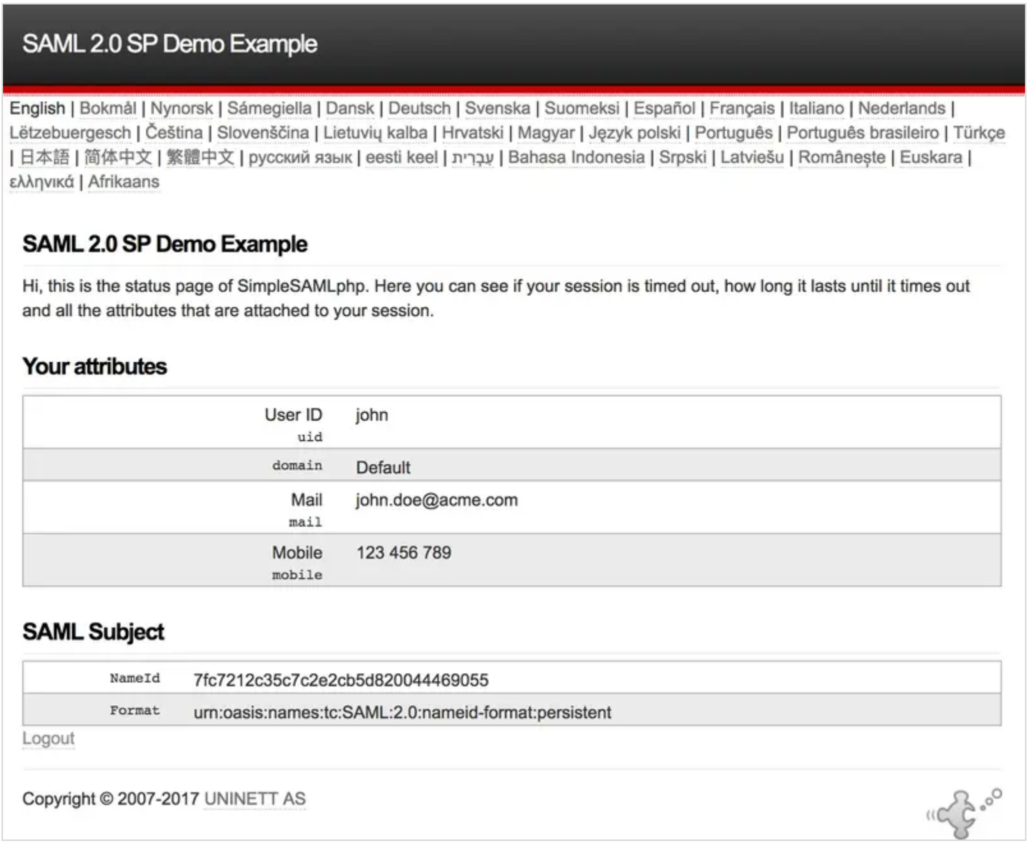


OpenID & SAML Provider

Redirecting to **sp.local** in 1 seconds.
Click the icon if your browser does not auto-redirect.

Provided by RCDevs Security Solutions

It’s done, we are authenticated:



SAML 2.0 SP Demo Example

English | Bokmål | Nynorsk | Sámegiella | Dansk | Deutsch | Svenska | Suomeksi | Español | Français | Italiano | Nederlands | Lëtzebuergesch | Čeština | Slovenščina | Lietuvių kalba | Hrvatski | Magyar | Język polski | Português | Português brasileiro | Türkçe | 日本語 | 简体中文 | 繁體中文 | русский язык | eesti keel | עברית | Bahasa Indonesia | Srpski | Latviešu | Românește | Euskara | ελληνικά | Afrikaans

SAML 2.0 SP Demo Example

Hi, this is the status page of SimpleSAMLphp. Here you can see if your session is timed out, how long it lasts until it times out and all the attributes that are attached to your session.

Your attributes

User ID uid	john
domain	Default
Mail mail	john.doe@acme.com
Mobile mobile	123 456 789

SAML Subject

NameId	7fc7212c35c7c2e2cb5d820044469055
Format	urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

Logout

Copyright © 2007-2017 UNINETT AS

We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-21 11:16:31] [192.168.1.220] [OpenID:Y84I9XHY] User not authenticated (entering login form)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] New login request (OpenOTP)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Username: john
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Domain: Default
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > ANY Password: xxxxxxx
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Sending openotpSimpleLogin request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] New openotpSimpleLogin SOAP request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Username: john
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Domain: Default
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Password: xxxxxxx
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Client ID: OpenID
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Source IP: 192.168.1.220
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Registered openotpSimpleLogin request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Resolved LDAP user: cn=john,o=Root
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Started transaction lock for user
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user mobiles: 123 456 789
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user emails: john.doe@acme.com
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTimeou
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=

[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user data: LoginCount
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Requested login factors: LDAP
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] LDAP password Ok
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Updated user data
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Sent success response
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] OpenOTP authentication success
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Resolved LDAP user: cn=john,o=Root
(cached)
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Login session started for cn=john,o=Root
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Sent SAML success response
```

3.2.2 Nextcloud

This was tested with Nextcloud 18.

3.2.2.1 Requirements

As a requirement, you need to install two apps in the app section:

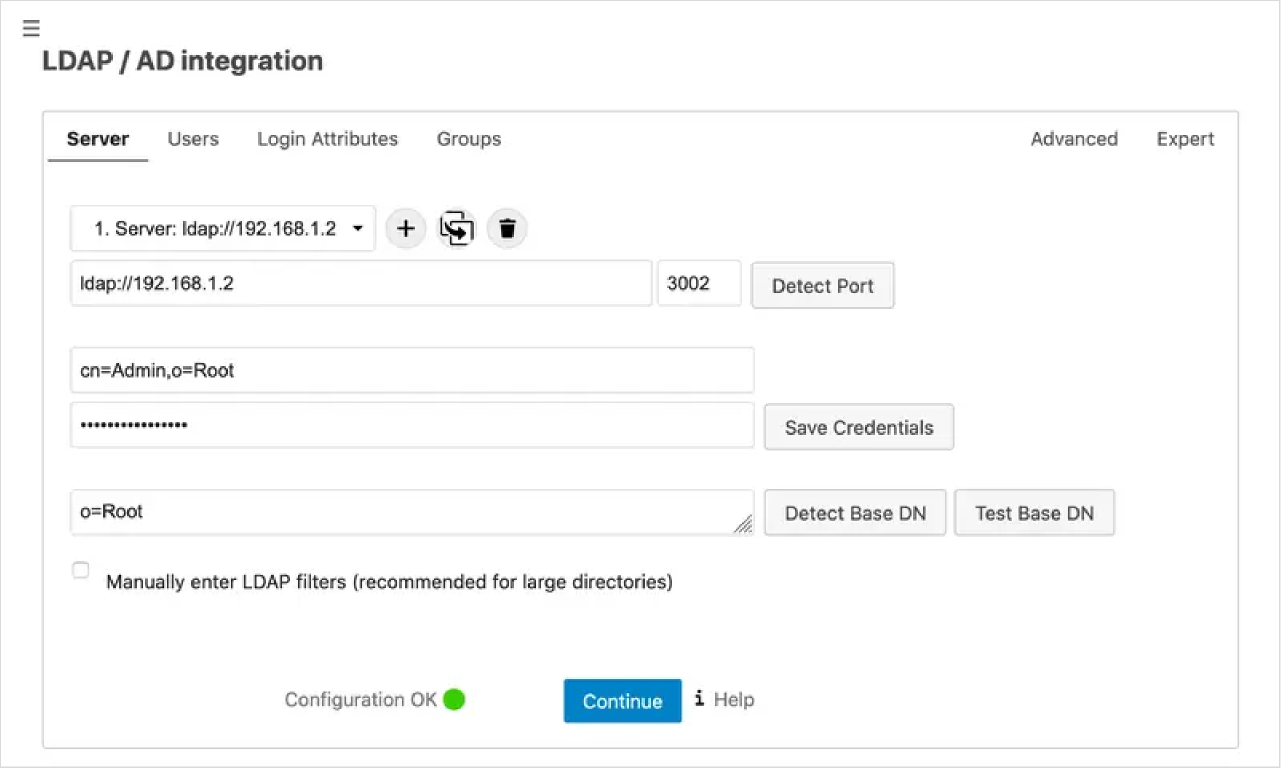
- LDAP user and group backend docs.nextcloud.com

- SSO & SAML authentication apps.nextcloud.com

3.2.2.2 Configuration of “LDAP / AD integration” app

Then, you need to configure first the LDAP app to synchronize users stored in your LDAP server.

First, configure the connection to the LDAP server. You can adapt what is showed in the screenshot. You should get a green Configuration OK when settings are well-defined.



LDAP / AD integration

1. Server: ldap://192.168.1.2

ldap://192.168.1.2 3002

cn=Admin,o=Root

••••••••••••••••

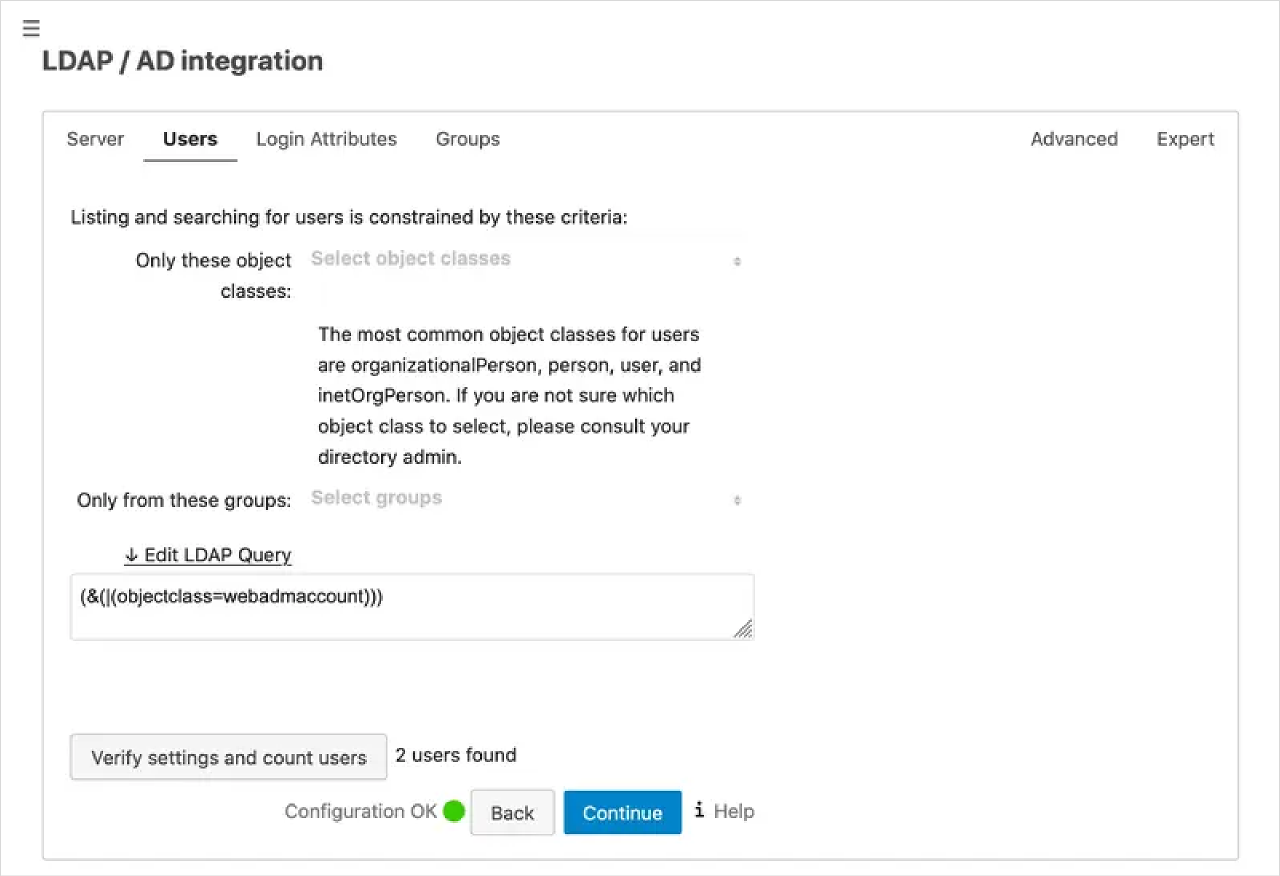
o=Root

□ Manually enter LDAP filters (recommended for large directories)

Configuration OK ●

Figure 3. LDAP / AD integration (server configuration)

Next, you can adapt the search query in order to get right users from the LDAP.



Listing and searching for users is constrained by these criteria:

Only these object classes: Select object classes

The most common object classes for users are organizationalPerson, person, user, and inetOrgPerson. If you are not sure which object class to select, please consult your directory admin.

Only from these groups: Select groups

↓ Edit LDAP Query

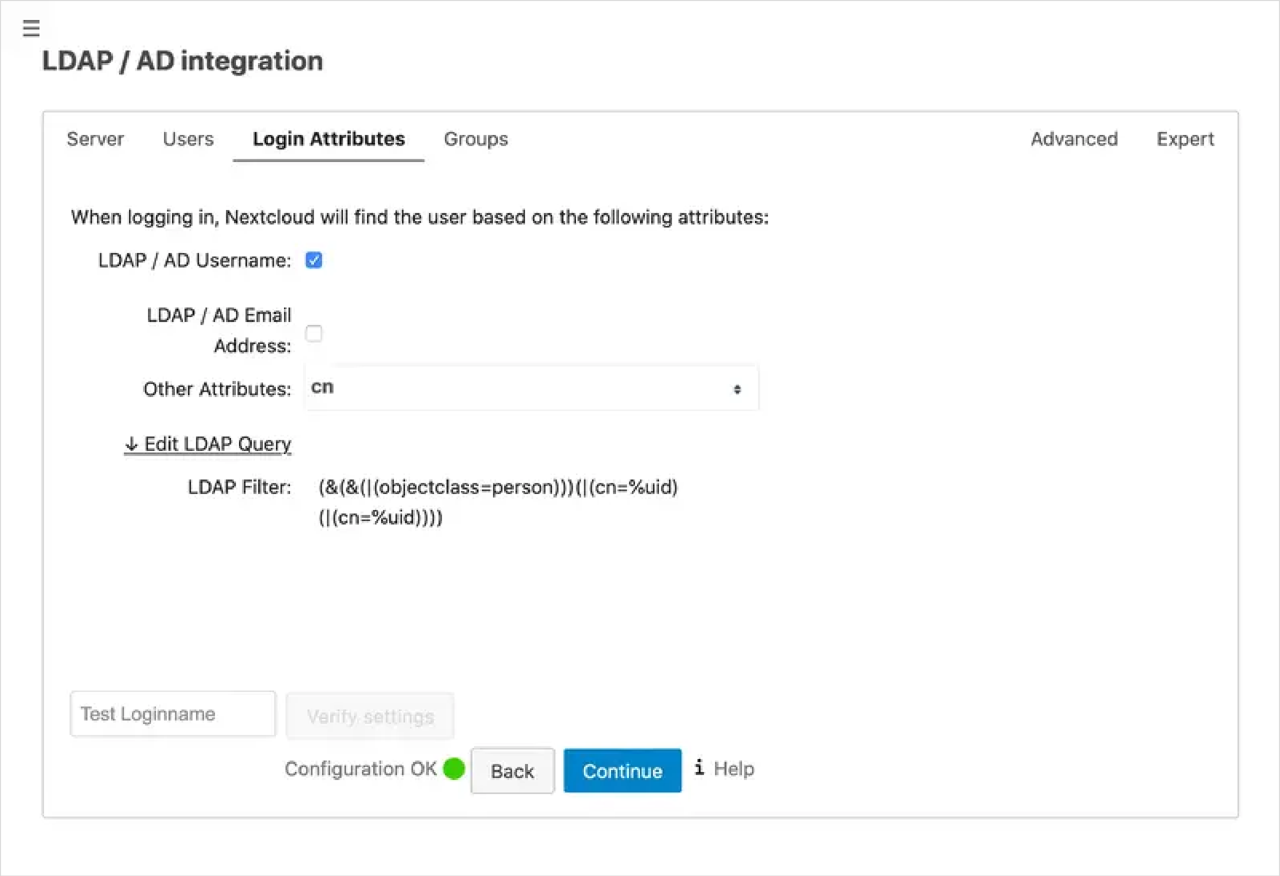
```
(&(|(objectclass=webadmaccount)))
```

2 users found

Configuration OK ●

Figure 4. LDAP / AD integration (user search query configuration)

Finally, configure the login attribute used to get the right username of users.



When logging in, Nextcloud will find the user based on the following attributes:

LDAP / AD Username: ☑

LDAP / AD Email Address: ☐

Other Attributes: cn

↓ Edit LDAP Query

LDAP Filter: (&(&(|(objectclass=person)))(|(cn=%uid)(|(cn=%uid))))

Configuration OK ●

Figure 5. LDAP / AD integration (Login attribute configuration)

3.2.2.3 Configuration of “SSO & SAML authentication” app

3.2.2.4 Global Settings

On “Global Settings”, it is only required to tick “Allow the use of multiple user back-ends (e.g. LDAP)”, so IdP login initiation can work (See 2.1.2.4). If you still need to authenticate using a local account of Nextcloud, you can use the following URL to access the direct login mode: `https://yournextcloudserver/login?direct=1`

3.2.2.5 General

In the General section, you can set the following elements:

- Attribute to map the UID to. setting;
- Optional display name of the identity provider (default: “SSO & SAML log in”) setting.

3.2.2.6 Identity Provider Data

In the Identity Provider Data section, you have to set the following elements:

- Identifier of the IdP entity (must be a URI);
- URL Target of the IdP where the SP will send the Authentication Request Message;
- URL Location of the IdP where the SP will send the SLO Request. For these three first settings, you need to set the URL of root of openid (e.g. `https://yournextcloudserver/webapps/openid/`).

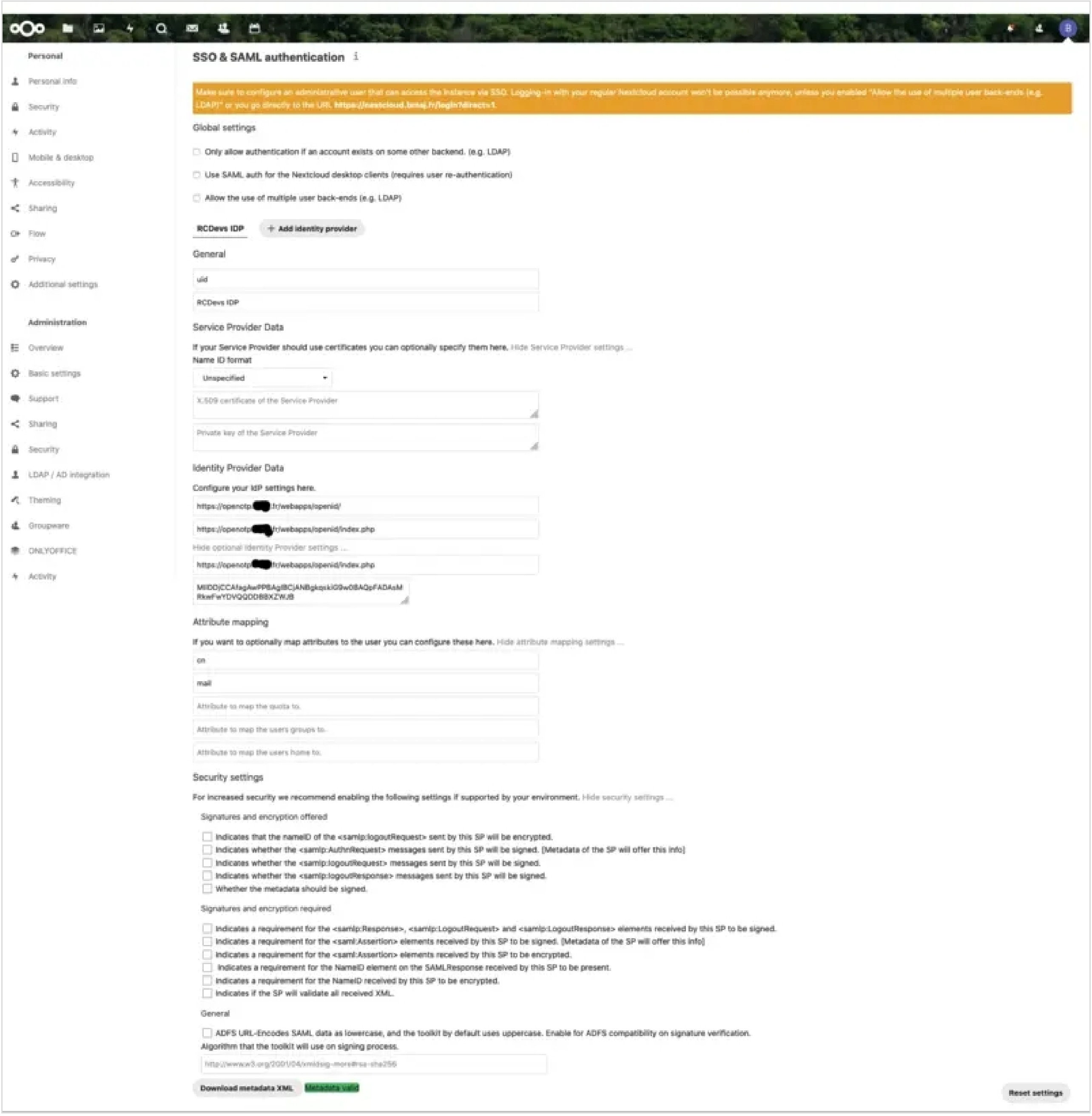
In order to set the Public X.509 certificate of the IdP setting, you can open saml URL (e.g. `https://yournextcloudserver/ws/saml/`) and copy and paste value contained in X509Certificate anchor.

3.2.2.7 Attribute mapping

Attribute mapping elements can also be set. Here, you can modify the following:

- Attribute to map the displayname to;
- Attribute to map the email address to;
- Attribute to map the quota to;

- Attribute to map the users groups to;
- Attribute to map the users home to;



SSO & SAML authentication

Make sure to configure an administrative user that can access the instance via SSO. Logging-in with your regular Nextcloud account won't be possible anymore, unless you enabled "Allow the use of multiple user back-ends (e.g. LDAP)" or you go directly to the URL https://nextcloud.bmaj.fr/login?direct=1.

Global settings

□ Only allow authentication if an account exists on some other backend. (e.g. LDAP)

□ Use SAML auth for the Nextcloud desktop clients (requires user re-authentication)

□ Allow the use of multiple user back-ends (e.g. LDAP)

RCDevs IDP | + Add identity provider

General

uid

RCDevs IDP

Service Provider Data

If your Service Provider should use certificates you can optionally specify them here. Hide Service Provider settings ...

Name ID format

Unspecified

X.509 certificate of the Service Provider

Private key of the Service Provider

Identity Provider Data

Configure your IdP settings here.

https://openotp[redacted].fr/webapps/openid/

https://openotp[redacted].fr/webapps/openid/index.php

Hide optional Identity Provider settings ...

https://openotp[redacted].fr/webapps/openid/index.php

MIIDDjCCAfagAwFPBAgIBCjANBgkqskiG9w0BAQpFADAsM
RkwFwYDVQQDDBBXZWJB

Attribute mapping

If you want to optionally map attributes to the user you can configure these here. Hide attribute mapping settings ...

cn

mail

Attribute to map the quota to.

Attribute to map the users groups to.

Attribute to map the users home to.

Security settings

For increased security we recommend enabling the following settings if supported by your environment. Hide security settings ...

Signatures and encryption offered

□ Indicates that the nameID of the <samlp:logoutRequest> sent by this SP will be encrypted.
□ Indicates whether the <samlp:AuthnRequest> messages sent by this SP will be signed. [Metadata of the SP will offer this info]
□ Indicates whether the <samlp:logoutRequest> messages sent by this SP will be signed.
□ Indicates whether the <samlp:logoutResponse> messages sent by this SP will be signed.
□ Whether the metadata should be signed.

Signatures and encryption required

□ Indicates a requirement for the <samlp:Response>, <samlp:LogoutRequest> and <samlp:LogoutResponse> elements received by this SP to be signed.
□ Indicates a requirement for the <saml:Assertion> elements received by this SP to be signed. [Metadata of the SP will offer this info]
□ Indicates a requirement for the <saml:Assertion> elements received by this SP to be encrypted.
□ Indicates a requirement for the NameID element on the SAMLResponse received by this SP to be present.
□ Indicates a requirement for the NameID received by this SP to be encrypted.
□ Indicates if the SP will validate all received XML.

General

□ ADFS URL-Encodes SAML data as lowercase, and the toolkit by default uses uppercase. Enable for ADFS compatibility on signature verification.

Algorithm that the toolkit will use on signing process.

http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

Figure 6. SSO & SAML authentication (openid configuration)

3.3 Other examples (OpenID/SAML)

3.3.1 Apache Guacamole

First you need to install the OpenID extension to Apache Guacamole. See Guacamole documentation for instructions.

Please note that the authentication extensions in the GUACAMOLE_HOME/extensions directory are loaded in alphabetical order, so if you have another authentication extension which is alphabetically before the OpenID extension, then the OpenID extension will not be loaded. This is the case for example with guacamole-auth-jdbc-mysql extension. To bypass this issue you can rename the guacamole-auth-openid-1.0.0.jar to for example guacamole-auth-0penid-1.0.0.jar.

Once the extension is installed, you can configure the OpenID settings in GUACAMOLE_HOME/guacamole.properties

```
#OpenID authentication
openid-authorization-endpoint: https://<openotp_server_address>/openid/index.php
openid-jwks-endpoint: https://<openotp_server_address>/openid/certs.php
openid-issuer: https://<openotp_server_address>/webapps/openid/
openid-client-id: Guacamole
openid-redirect-uri: https://<guacamole_server_address>/guacamole/
```

Once the configuration is completed, you need to restart tomcat for it to take effect. If you want to log in as an existing Guacamole Admin user (for example guacadmin) while OpenID is enabled, you need to create that user in WebADM as well.

3.3.2 GitLab

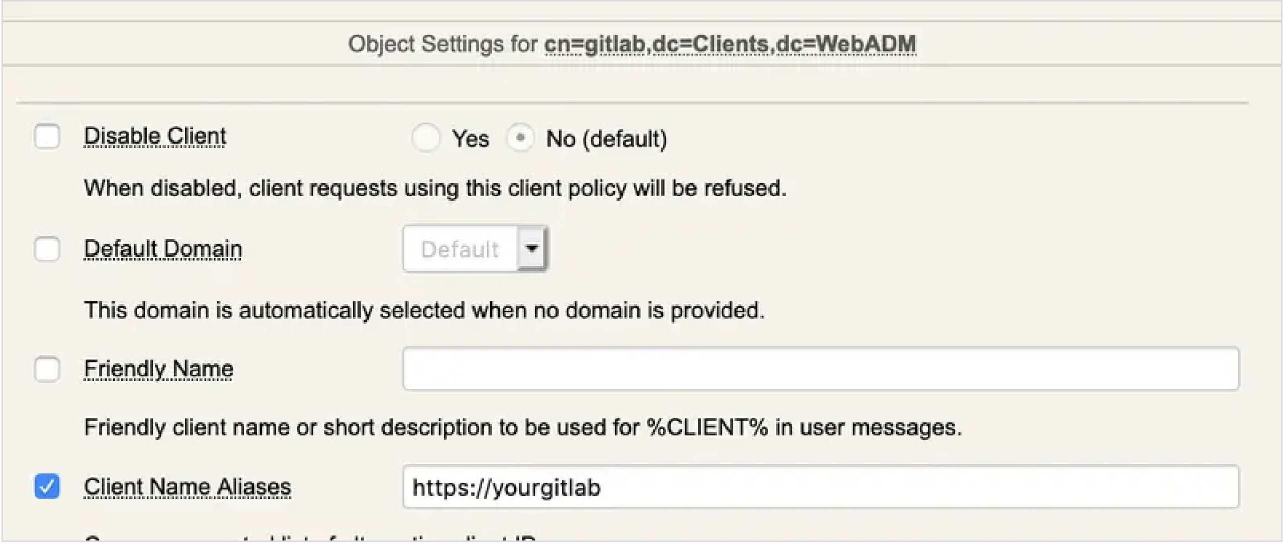
This was tested with GitLab Enterprise Edition 13.2.1.

3.3.2.1 Requirements

The following LDAP attributes must be returned to SAML assertions to GitLab:

- first_name=givenname
- last_name=sn
- mail=mail

It is recommended to add this OpenID setting in a client policy specific to your GitLab instance. First create a client policy (you can name it GitLab) and put the client ID provided by GitLab (this can be found in the webadm.log file) in the “Client Name Aliases” setting:



Object Settings for cn=gitlab,dc=Clients,dc=WebADM

☐ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☐ Default Domain Default

This domain is automatically selected when no domain is provided.

☐ Friendly Name

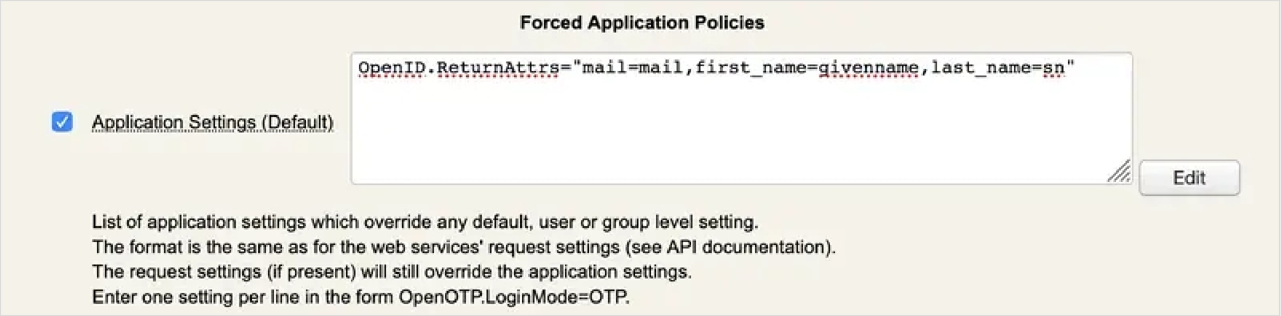
Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases https://yourgitlab

Figure 1. GitLab (client policy configuration)

Next, still on the client policy, add to the “Forced Application Policies” setting the following to properly configure the returned attributes for the SAML assertion:

OpenID.ReturnAttrs="mail=mail,first_name=givenname,last_name=sn"



Forced Application Policies

☑ Application Settings (Default)

```
OpenID.ReturnAttrs="mail=mail,first_name=givenname,last_name=sn"
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Figure 2. GitLab (client policy configuration)

3.3.2.2 Configuring SSO in GitLab

3.3.2.2.1 Enable SSO

First you need to enable SSO, and to permit auto creation of users.

You can add these lines for an Omnibus package installation to `config/gitlab.yml` file:

```
gitlab_rails['omniauth_allow_single_sign_on'] = ['saml']
gitlab_rails['omniauth_block_auto_created_users'] = false
gitlab_rails['omniauth_auto_link_saml_user'] = true
```

You can add these lines for a source installation to `config/gitlab.yml` file:

```
omniauth:
  enabled: true
  allow_single_sign_on: ["saml"]
  block_auto_created_users: false
  auto_link_saml_user: true
```

3.3.2.2.2 Add WebADM IdP

Next, you have to add the configuration of your IdP, still in `config/gitlab.yml` file.

The following parameters must be configured properly:

- **assertion_consumer_service_url**: this must match the URL of your gitlab, appended with `/users/auth/saml/callback`
- **idp_cert_fingerprint**: this is the fingerprint of the certificate provided by the SAML of your openotp. It can be retrieved using this command:

```
curl -ks https://youropenotp/ws/saml | grep SHA1 | awk '{print $5}' |  sed 's/../&:/g;s/:$//'
```

- **idp_sso_target_url**: this must match the URL domain of your openotp, appended with `/webapps/openid/index.php`
- **issuer**: this must be a unique name which will be used by openotp to identify your GitLab.
- **label**: this is the link name displayed on the sign-page to do SSO.

For an Omnibus package installation, add the following and adapt to your needs:

```
gitlab_rails['omniauth_providers'] = [
  {
    name: 'saml',
    args: {
             assertion_consumer_service_url: 'https://yourgitlab/users/auth/saml/callback',
             idp_cert_fingerprint: '43:51:43:a1:b5:fc:8b:b7:0a:3a:a9:b1:0f:66:73:a8',
             idp_sso_target_url: 'https://youropenotp/webapps/openid/index.php',
             issuer: 'https://yourgitlab',
             name_identifier_format: 'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent'
           },
    label: 'Company Login' # optional label for SAML login button, defaults to "Saml"
  }
]
```

For a source installation, add the following and adapt to your needs:

```
omniauth:
  providers:
    - {
        name: 'saml',
        args: {
               assertion_consumer_service_url: 'https://gitlab.example.com/users/auth/saml/callback',
               idp_cert_fingerprint: '43:51:43:a1:b5:fc:8b:b7:0a:3a:a9:b1:0f:66:73:a8',
               idp_sso_target_url: 'https://youropenotp/webapps/openid/index.php',
               issuer: 'https://yourgitlab',
               name_identifier_format: 'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent'
             },
        label: 'Company Login' # optional label for SAML login button, defaults to "Saml"
      }
```

3.3.3 Grafana

First, create a new or update an existing Client Policy in WebADM > Admin > Client Policies. The policy name or friendly name must match the client_id defined in Grafana configuration (see below).

In the client policy, configure Application Settings > Edit > OpenID & SAML Provider > Client Secret. This secret must match the client_secret defined in Grafana.

Once these settings are applied, you can configure Grafana to use OpenOTP IdP for SSO login:

```
[auth.generic_oauth]
enabled = true
name = OpenOTP
allow_sign_up = true
client_id = grafana
client_secret = secret
scopes = openid profile email
auth_url = https://<openotp_server_address>/webapps/openid/index.php
token_url = https://<openotp_server_address>/webapps/openid/index.php
api_url = https://<openotp_server_address>/webapps/openid/index.php
tls_skip_verify_insecure = true
```

3.3.4 OnlyOffice

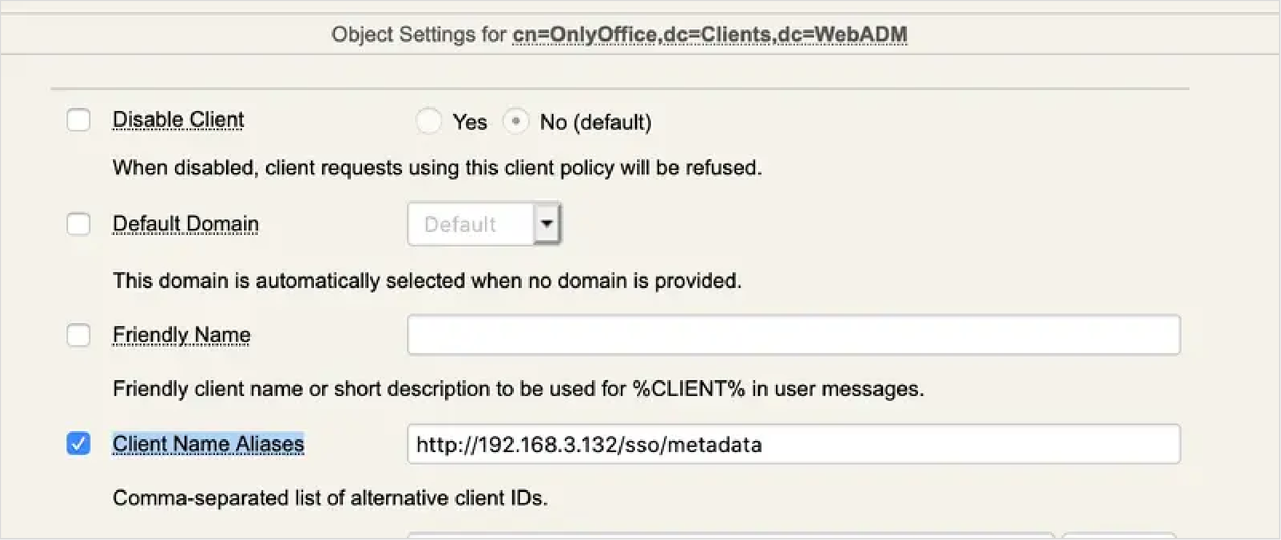
This was tested with OnlyOffice Enterprise Edition 10.5.3.

3.3.4.1 Requirements

The following LDAP attributes must be returned to SAML assertions to OnlyOffice (Location, Title, and Phone are optional attributes):

- givenName=givenname
- sn=sn
- mail=mail

It is recommended to add this OpenID setting in a client policy specific to your OnlyOffice instance. First create a client policy (you can name it OnlyOffice) and put the client ID provided by OnlyOffice (this can be found in the webadm.log file) in the “Client Name Aliases” setting:



Object Settings for cn=OnlyOffice,dc=Clients,dc=WebADM

□ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

□ Default Domain: Default

This domain is automatically selected when no domain is provided.

□ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

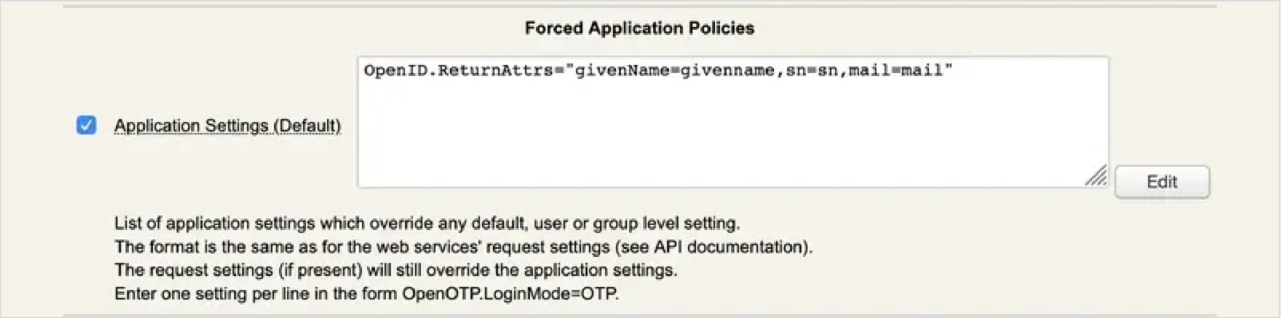
☑ Client Name Aliases: http://192.168.3.132/sso/metadata

Comma-separated list of alternative client IDs.

Figure 7. OnlyOffice (client policy configuration)

Next, still on the client policy, add to the “Forced Application Policies” setting the following to properly configure the returned attributes for the SAML assertion:

OpenID.ReturnAttrs=“givenName=givenname,sn=sn,mail=mail”



Forced Application Policies

☑ Application Settings (Default)

```
OpenID.ReturnAttrs="givenName=givenname,sn=sn,mail=mail"
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Figure 8. OnlyOffice (client policy configuration)

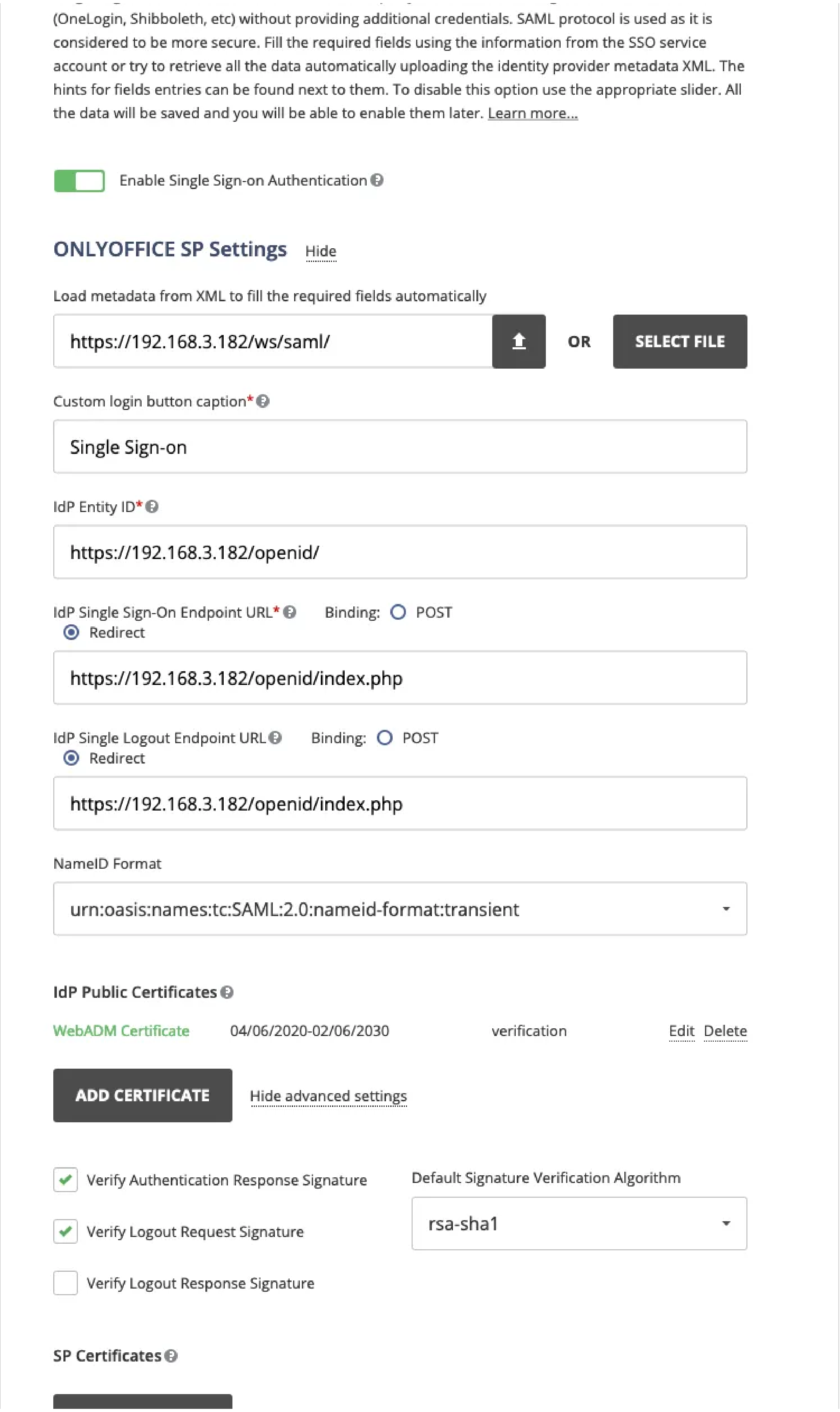
3.3.4.2 Configuring SSO in OnlyOffice

Open the following URL of your OnlyOffice: `https://youronlyoffice/controlpanel/sso`

Enable SSO, put the URL of your webadm (or waproxy if you have deployed one) in the “URL to IdP Metadata XML” field, and click on Load data button. This will pre-fill other input settings. You can click on the save button.

SSO

Single Sign-on allows to enable or disable third-party authentication using the installed SSO services



(OneLogin, Shibboleth, etc) without providing additional credentials. SAML protocol is used as it is considered to be more secure. Fill the required fields using the information from the SSO service account or try to retrieve all the data automatically uploading the identity provider metadata XML. The hints for fields entries can be found next to them. To disable this option use the appropriate slider. All the data will be saved and you will be able to enable them later. Learn more...

Enable Single Sign-on Authentication

ONLYOFFICE SP Settings Hide

Load metadata from XML to fill the required fields automatically

https://192.168.3.182/ws/saml/

OR

SELECT FILE

Custom login button caption*

Single Sign-on

IdP Entity ID*

https://192.168.3.182/openid/

IdP Single Sign-On Endpoint URL* Binding: ○ POST ● Redirect

https://192.168.3.182/openid/index.php

IdP Single Logout Endpoint URL Binding: ○ POST ● Redirect

https://192.168.3.182/openid/index.php

NameID Format

urn:oasis:names:tc:SAML:2.0:nameid-format:transient

IdP Public Certificates

WebADM Certificate	04/06/2020-02/06/2030	verification	Edit Delete

ADD CERTIFICATE Hide advanced settings

☑ Verify Authentication Response Signature

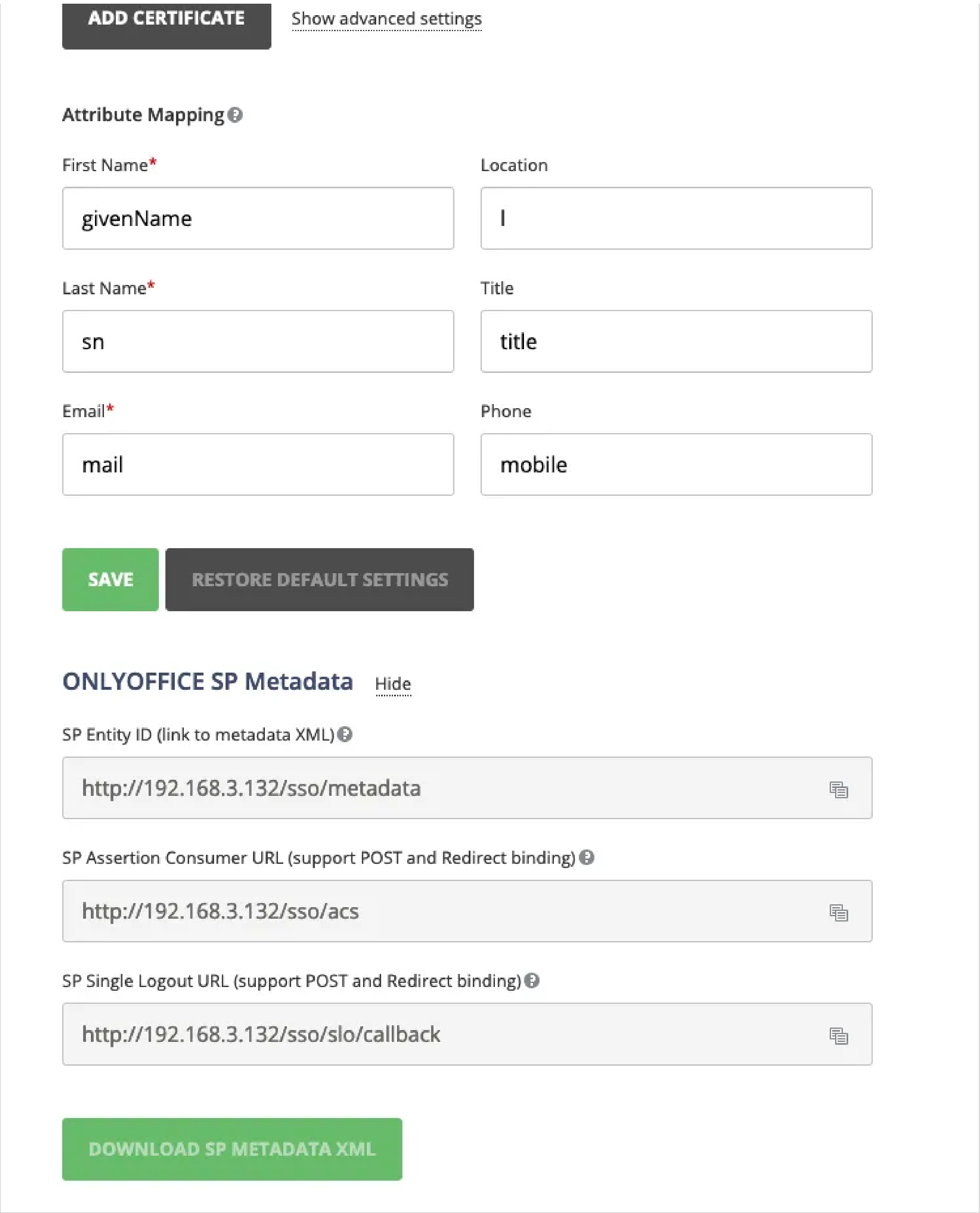
☑ Verify Logout Request Signature

□ Verify Logout Response Signature

Default Signature Verification Algorithm

rsa-sha1

SP Certificates



Attribute Mapping

First Name*	givenName
Location	l
Last Name*	sn
Title	title
Email*	mail
Phone	mobile

ONLYOFFICE SP Metadata

SP Entity ID (link to metadata XML): http://192.168.3.132/sso/metadata

SP Assertion Consumer URL (support POST and Redirect binding): http://192.168.3.132/sso/acs

SP Single Logout URL (support POST and Redirect binding): http://192.168.3.132/sso/slo/callback

Figure 9. OnlyOffice (SSO configuration)

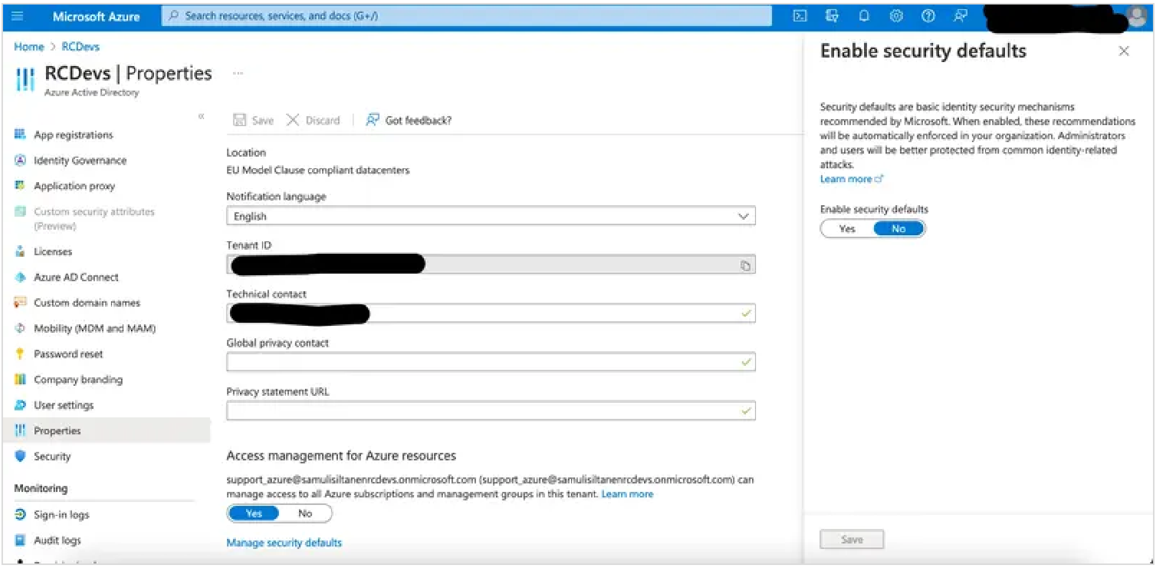
3.3.5 MS Office 365/Azure Integration with an Active Directory Backend

3.3.5.1 Prerequistes

- You need an Administrator on the AZURE AD,
- You need to install and configure Azure Sync on one of your Domain Controler,

- You need have a Windows PowerShell with the Azure AD PowerShell module installed,
- You need at least WebADM 2.0.16 and OpenID 1.4.11 versions.

> ### ⚠ Important Note
>
> We noticed that if “Default Security policies” are enabled on Azure Active Directory, Azure is expecting an MFA login to access Azure resources. This policy must be disabled else, the redirection to Azure/Office 365 after the authentiation on WebADM IDP will failed because Azure didn’t know that the MFA has been played with OpenOTP. There is maybe the possibility to customize this default policy on Azure to avoid this behavior and the expected 2FA. Please refer to Azure documentation for that part. On our side, we just disabled it. Refer to the screeshot below.



[redacted]

RCDevs | Properties
Azure Active Directory

Location
EU Model Clause compliant datacenters

Notification language
English

Tenant ID
[redacted]

Technical contact
[redacted]

Global privacy contact

Privacy statement URL

Access management for Azure resources

support_azure@samulisiltanenrcdevs.onmicrosoft.com (support_azure@samulisiltanenrcdevs.onmicrosoft.com) can manage access to all Azure subscriptions and management groups in this tenant. Learn more

[✓ Yes] No

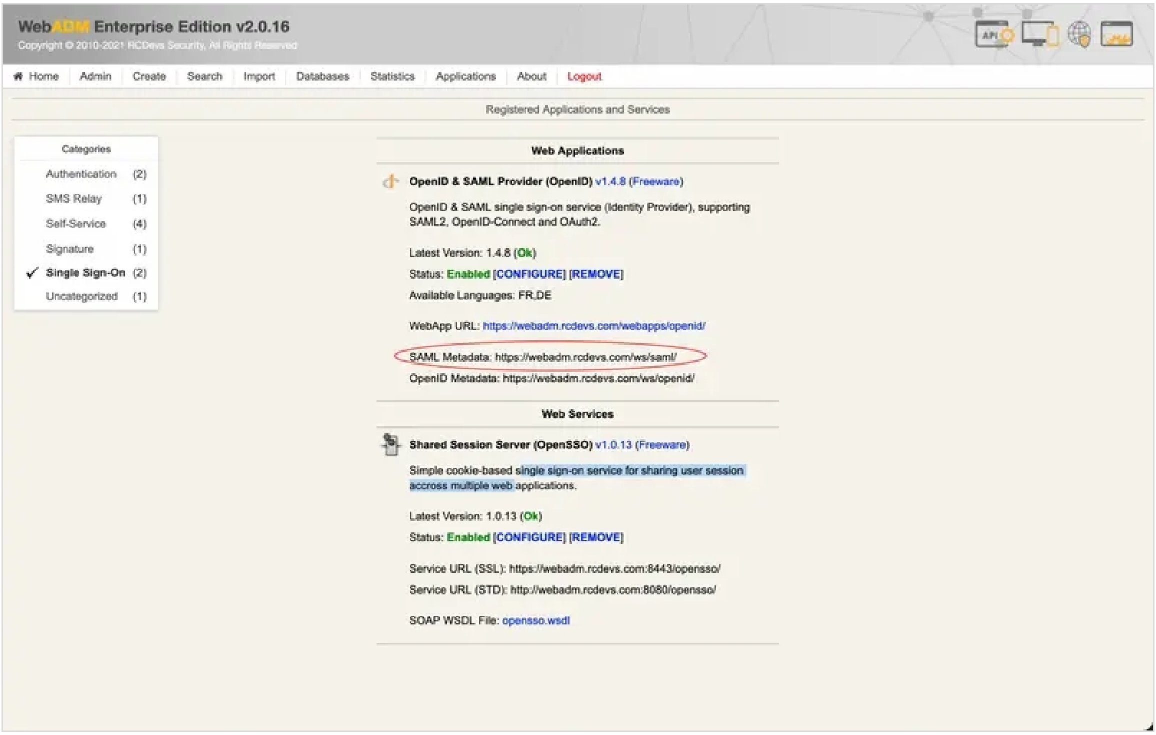
Enable security defaults

Security defaults are basic identity security mechanisms recommended by Microsoft. When enabled, these recommendations will be automatically enforced in your organization. Administrators and users will be better protected from common identity-related attacks.
Learn more

Enable security defaults
Yes [✓ No]

3.3.5.2 Get your configuration of your IDP on WebADM

You will need for the next step Log on your webadm and go to Applications > Single Sign-On and check the link SAML Metadata



WebADM Enterprise Edition v2.0.16
Copyright © 2010-2021 RCDevs Security, All Rights Reserved
Home
Admin
Create
Search
Import
Databases
Statistics
Applications
About
Logout

Registered Applications and Services

Categories	
Authentication	(2)
SMS Relay	(1)
Self-Service	(4)
Signature	(1)
✓ Single Sign-On	(2)
Uncategorized	(1)

Web Applications

OpenID & SAML Provider (OpenID) v1.4.8 (Freeware)

OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.4.8 (Ok)
Status: Enabled [CONFIGURE] [REMOVE]
Available Languages: FR,DE

WebApp URL: https://webadm.rcdevs.com/webapps/openid/

SAML Metadata: https://webadm.rcdevs.com/ws/saml/
OpenID Metadata: https://webadm.rcdevs.com/ws/openid/

Web Services

Shared Session Server (OpenSSO) v1.0.13 (Freeware)

Simple cookie-based single sign-on service for sharing user session accross multiple web applications.

Latest Version: 1.0.13 (Ok)
Status: Enabled [CONFIGURE] [REMOVE]

Service URL (SSL): https://webadm.rcdevs.com:8443/opensso/
Service URL (STD): http://webadm.rcdevs.com:8080/opensso/

SOAP WSDL File: opensso.wsdl

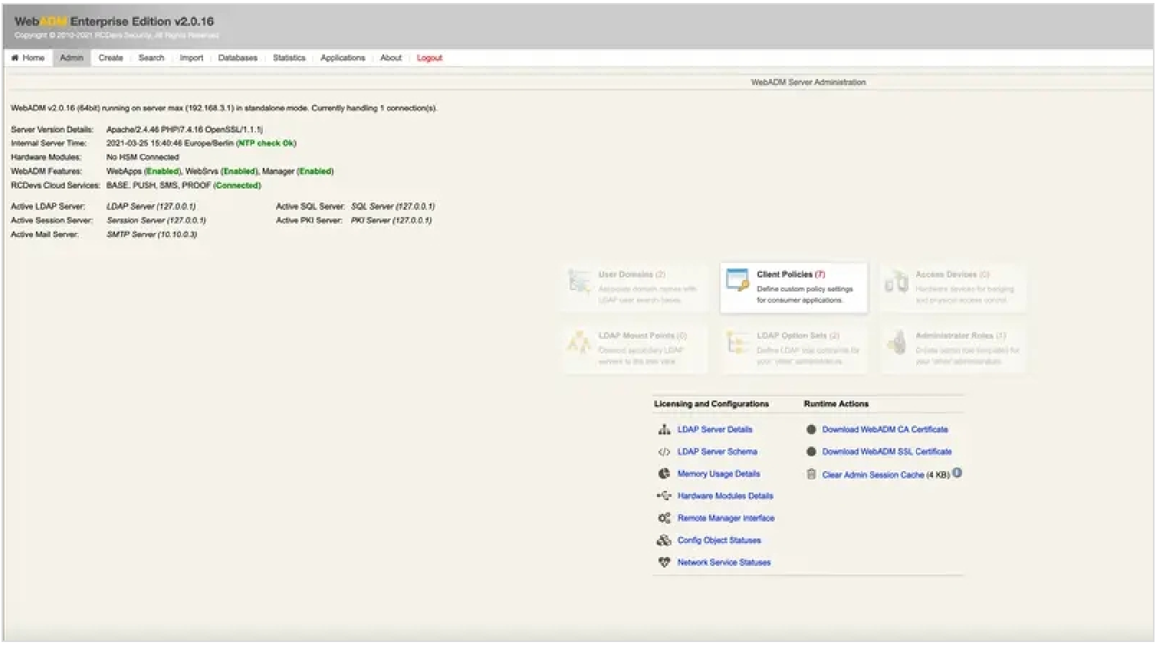
Figure 3.4.2.1 get your SAML Metadata on WebAdm

Open the link in a browser In the XML File you need to get the:

- entityID (https://webadm.foo.bar/)
- X509Certificate (XXXXXXX-X509Certificate-XXXXXXXX)
- SingleSignOnService location (https://webadm.foo.bar/webapps/openid/index.php)

3.3.5.3 Configure properly your IDP and your Policy on webadm

From `WebADM Admin GUI`, click on `Admin` tab, click on `Client Policy` box and go down to click on `Add Client`.



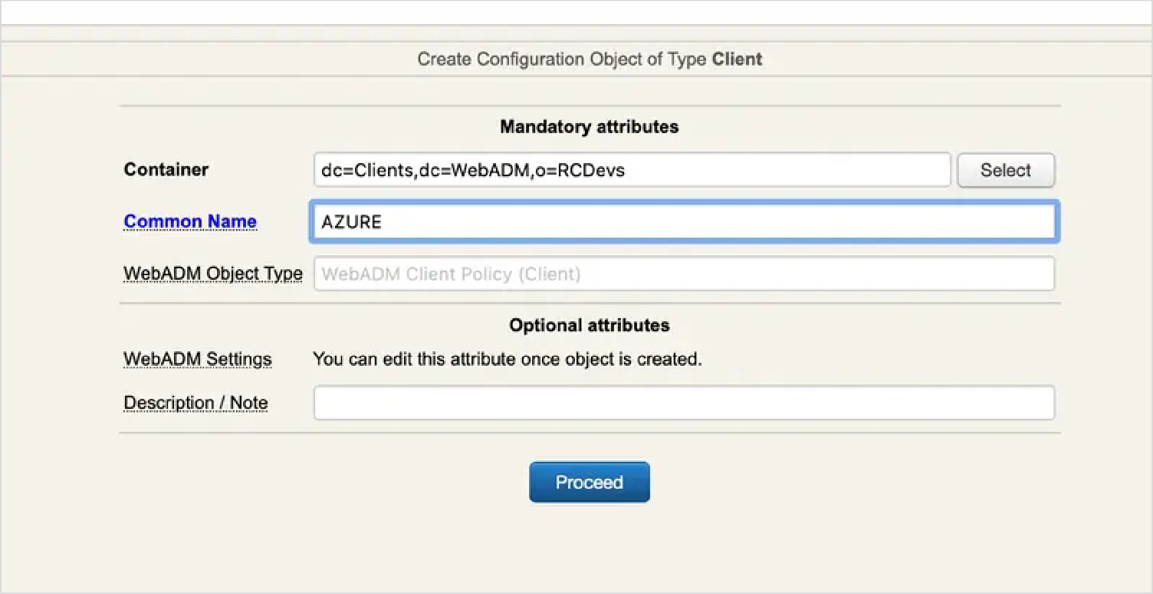
WebADM v2.0.16 (64bit) running on server max (192.168.3.1) in standalone mode. Currently handling 1 connection(s).

Server Version Details:	Apache/2.4.46 PHP/7.4.16 OpenSSL/1.1.1j
Internal Server Time:	2021-03-25 15:40:46 Europe/Berlin (NTP check Ok)
Hardware Modules:	No HSM Connected
WebADM Features:	WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services:	BASE, PUSH, SMS, PROOF (Connected)

Active LDAP Server:	*LDAP Server (127.0.0.1)*	Active SQL Server:	*SQL Server (127.0.0.1)*
Active Session Server:	*Session Server (127.0.0.1)*	Active PKI Server:	*PKI Server (127.0.0.1)*
Active Mail Server:	*SMTP Server (10.10.0.3)*		

Figure 3.4.3.1 Select Client Policy on WebADM

Give any name in Common Name to your Client Policy (here we use `AZURE`). Click `Proceed` then click on `Create Object`.



Mandatory attributes

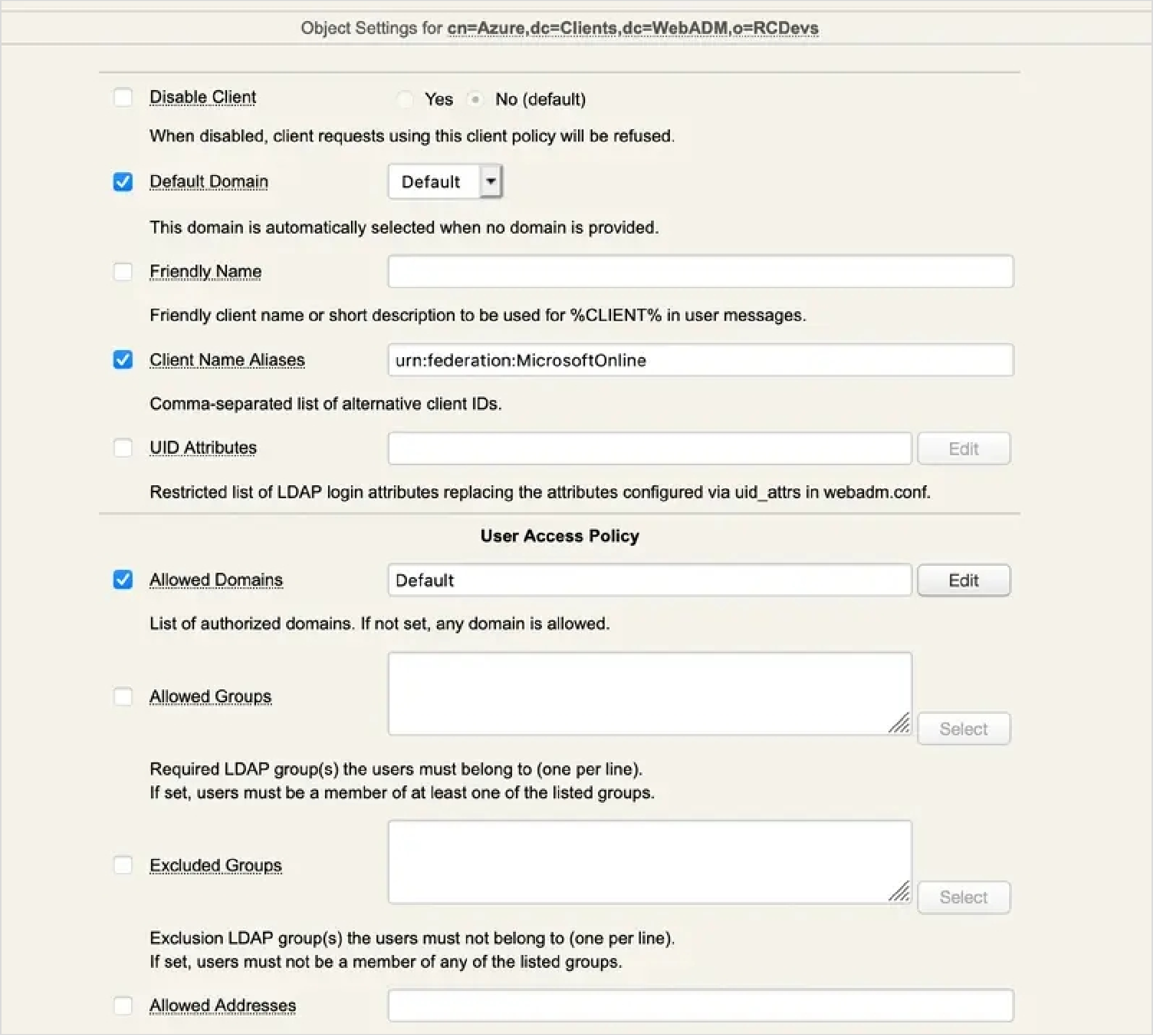
Container	dc=Clients,dc=WebADM,o=RCDevs
Common Name	AZURE
WebADM Object Type	WebADM Client Policy (Client)

Optional attributes

WebADM Settings	You can edit this attribute once object is created.
Description / Note	

Figure 3.4.3.2 Click on Add Client on WebADM

- Select your `Domain`
- Set your Client Name Aliases to: `urn:federation:MicrosoftOnline`



Object Settings for cn=Azure,dc=Clients,dc=WebADM,o=RCDevs

□ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☑ Default Domain: Default

This domain is automatically selected when no domain is provided.

□ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases: urn:federation:MicrosoftOnline

Comma-separated list of alternative client IDs.

□ UID Attributes [Edit]

Restricted list of LDAP login attributes replacing the attributes configured via uid_attrs in webadm.conf.

User Access Policy

☑ Allowed Domains: Default [Edit]

List of authorized domains. If not set, any domain is allowed.

□ Allowed Groups [Select]

Required LDAP group(s) the users must belong to (one per line).
If set, users must be a member of at least one of the listed groups.

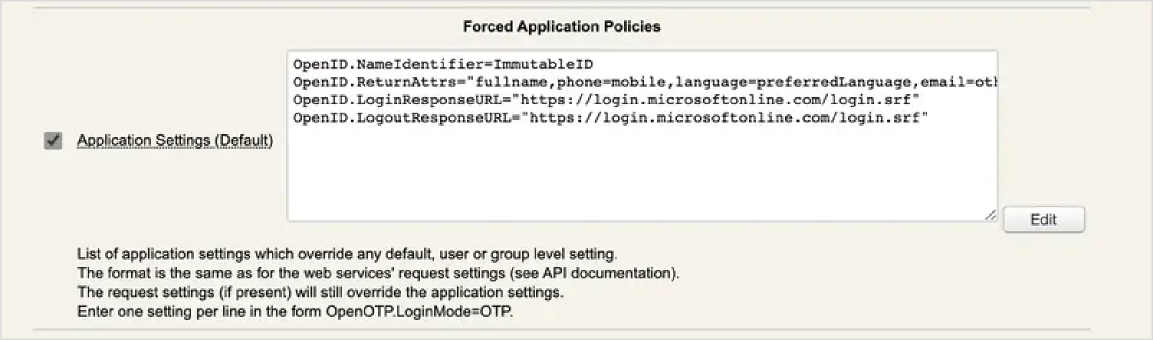
□ Excluded Groups [Select]

Exclusion LDAP group(s) the users must not belong to (one per line).
If set, users must not be a member of any of the listed groups.

□ Allowed Addresses

Figure 3.4.3.2 Select your Default Domain in WebAdm

Then click `EDIT` on Application Settings (Default)



Forced Application Policies

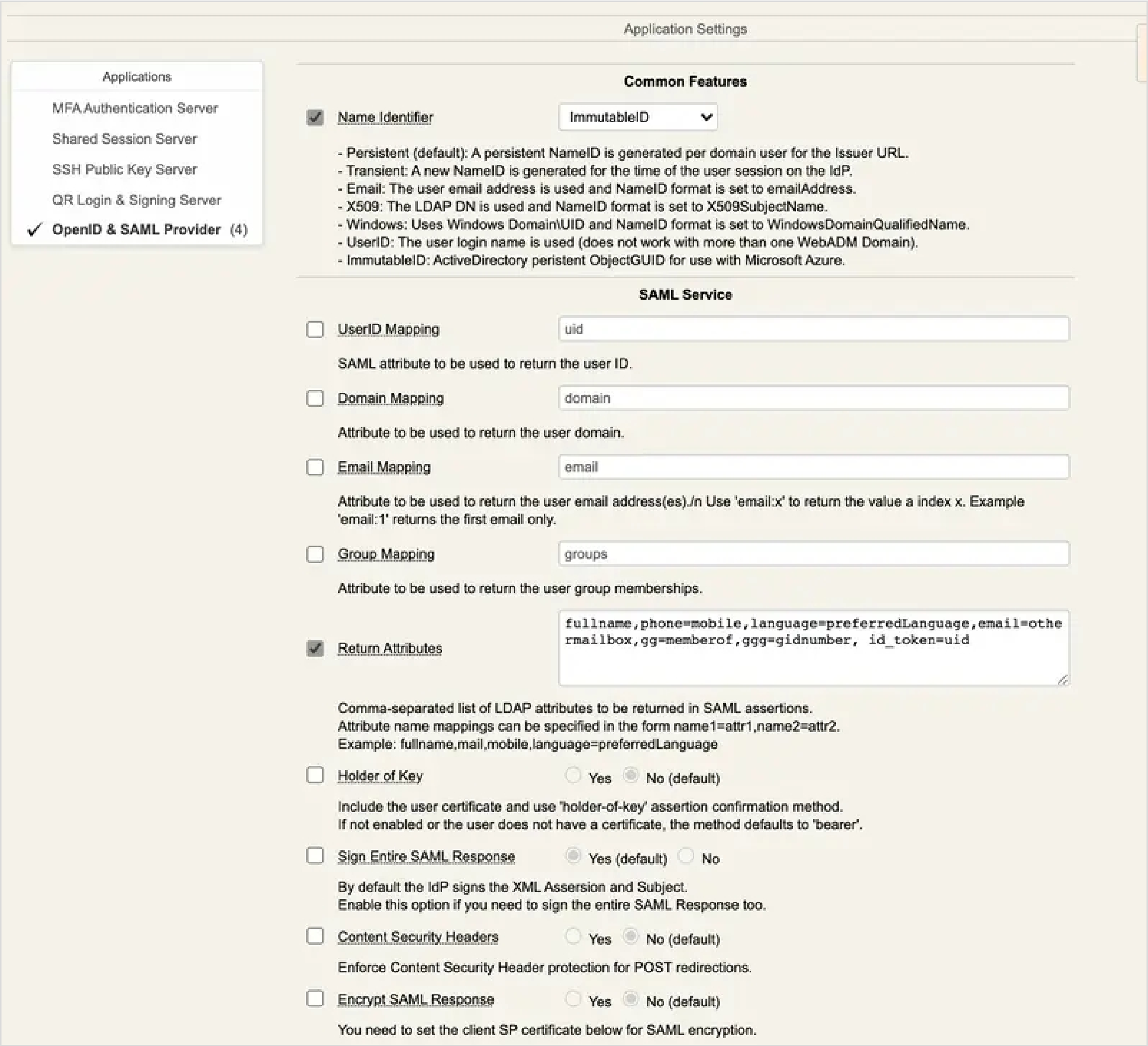
☑ Application Settings (Default) [Edit]

```
OpenID.NameIdentifier=ImmutableID
OpenID.ReturnAttrs="fullname,phone=mobile,language=preferredLanguage,email=ot
OpenID.LoginResponseURL="https://login.microsoftonline.com/login.srf"
OpenID.LogoutResponseURL="https://login.microsoftonline.com/login.srf"
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Figure 3.4.3.3 Click EDIT on Application Settings in WebAdm

- Set Name Identifier to `ImmutableID`
- Set Return Attributes you want to retun in the SAML assertion like `fullname,phone=mobile,language=preferredLanguage,email=othermailbox`
- Set Assertion Consumer Service URL to `https://login.microsoftonline.com/login.srf`
- Set Logout Consumer Service URL to `https://login.microsoftonline.com/login.srf`



Application Settings

Applications

- MFA Authentication Server
- Shared Session Server
- SSH Public Key Server
- QR Login & Signing Server
- ✓ OpenID & SAML Provider (4)

Common Features

☑ Name Identifier: ImmutableID

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

SAML Service

☐ UserID Mapping: uid

SAML attribute to be used to return the user ID.

☐ Domain Mapping: domain

Attribute to be used to return the user domain.

☐ Email Mapping: email

Attribute to be used to return the user email address(es)./n Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

☐ Group Mapping: groups

Attribute to be used to return the user group memberships.

☑ Return Attributes:

```
fullname,phone=mobile,language=preferredLanguage,email=othe
rmailbox,gg=memberof,ggg=gidnumber, id_token=uid
```

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

☐ Holder of Key: ○ Yes ● No (default)

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

☐ Sign Entire SAML Response: ● Yes (default) ○ No

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

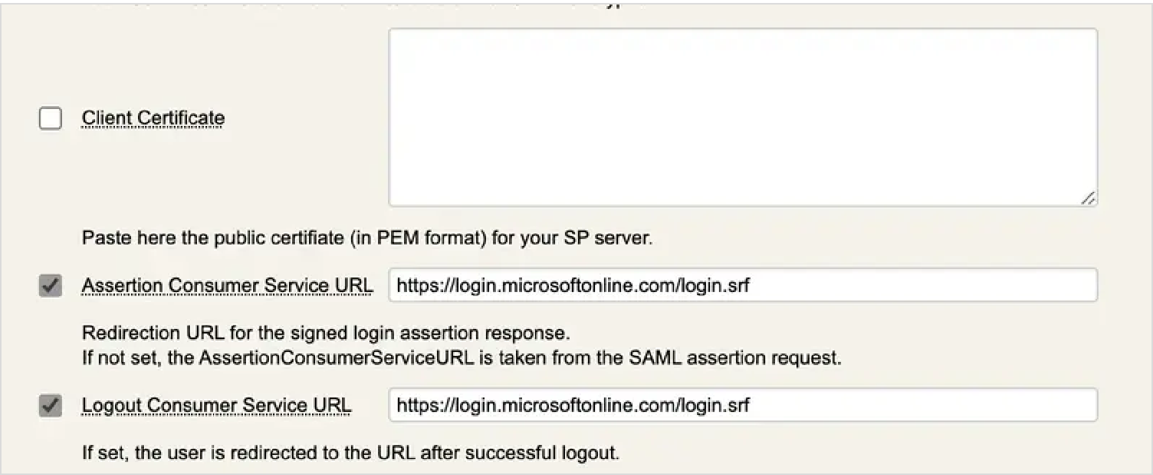
☐ Content Security Headers: ○ Yes ● No (default)

Enforce Content Security Header protection for POST redirections.

☐ Encrypt SAML Response: ○ Yes ● No (default)

You need to set the client SP certificate below for SAML encryption.

Figure 3.4.3.3 Set Name Identifier to Persistent in WebAdm



☐ Client Certificate

Paste here the public certifiate (in PEM format) for your SP server.

☑ Assertion Consumer Service URL: https://login.microsoftonline.com/login.srf

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

☑ Logout Consumer Service URL: https://login.microsoftonline.com/login.srf

If set, the user is redirected to the URL after successful logout.

Figure 3.4.3.4 Set Assertion and logout consumer service URLs

Click on `Apply`

Click Again on `Apply` and the configuration is done.

3.3.5.4 Set your OpenOTP IDP on AZURE with your domain

Lauch a Windows PowerShell. Connect to AZURE with your Administrator

```
PS C:\Users\admin> Connect-MsolService
```

You will need for the next step :

- entityID (https://webadm.foo.bar/)
- X509Certificate (XXXXXXX-X509Certificate-XXXXXXXX)
- SingleSignOnService location (https://webadm.foo.bar/webapps/openid/index.php)

Set the Federated authentification methode for your domain

```
PS C:\Users\admin> Set-MSolDomainAuthentication  -DomainName foo.bar -IssuerUri
https://webadm.foo.bar/ -FederationBrandName rcdevs.com -LogOffUri
https://webadm.foo.bar/webapps/openid/index.php -PassiveLogOnUri
https://webadm.foo.bar/webapps/openid/index.php -SigningCertificate XXXXXXX-X509Certificate-
XXXXXXXX -PreferredAuthenticationProtocol "SAMLP" -Authentication Federated
```

Now you should be able to log in the Azure page or on the Office 365 page. You can access to Azure of Office 365 login page, provide your email address or UPN. you should be redirected to the WebADM OpenID login page. Provide your credentials to login on the IDP. After a successful login on the IDP you will be redirected and logged into Azure or Office 365.

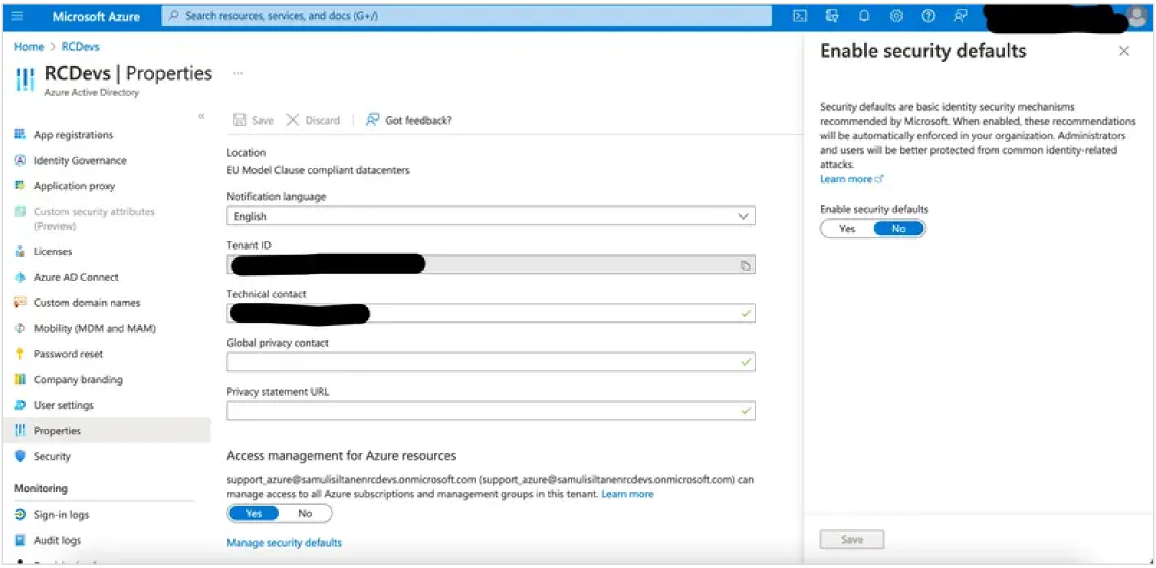
3.3.6 MS Office 365/Azure Integration without an Active Directory Backend

3.3.6.1 Prerequites

- You need to have a user Administrator on the AZURE AD
- You need to install on a Windows machine Connect-MsolService and New-MsolUser cmdlets,
- You need have a Windows PowerShell with the Azure AD PowerShell module installed,
- You need at least WebADM 2.0.16 and OpenID 1.4.11 versions.

⚠ Important Note

We noticed that if “Default Security policies” are enabled on Azure Active Directory, Azure is expecting an MFA login to access Azure resources. This policy must be disabled else, the redirection to Azure/Office 365 after the authentiation on WebADM IDP will failed because Azure didn’t know that the MFA has been played with OpenOTP. There is maybe the possibility to customize this default policy on Azure to avoid this behavior and the expected 2FA. Please refer to Azure documentation for that part. On our side, we just disabled it. Refer to the screeshot below.



Location
EU Model Clause compliant datacenters

Notification language
English

Tenant ID
[redacted]

Technical contact
[redacted]

Global privacy contact

Privacy statement URL

Access management for Azure resources

support_azure@samulisiltanenrcdevs.onmicrosoft.com (support_azure@samulisiltanenrcdevs.onmicrosoft.com) can manage access to all Azure subscriptions and management groups in this tenant. Learn more

[✓ Yes] No

Manage security defaults

Enable security defaults

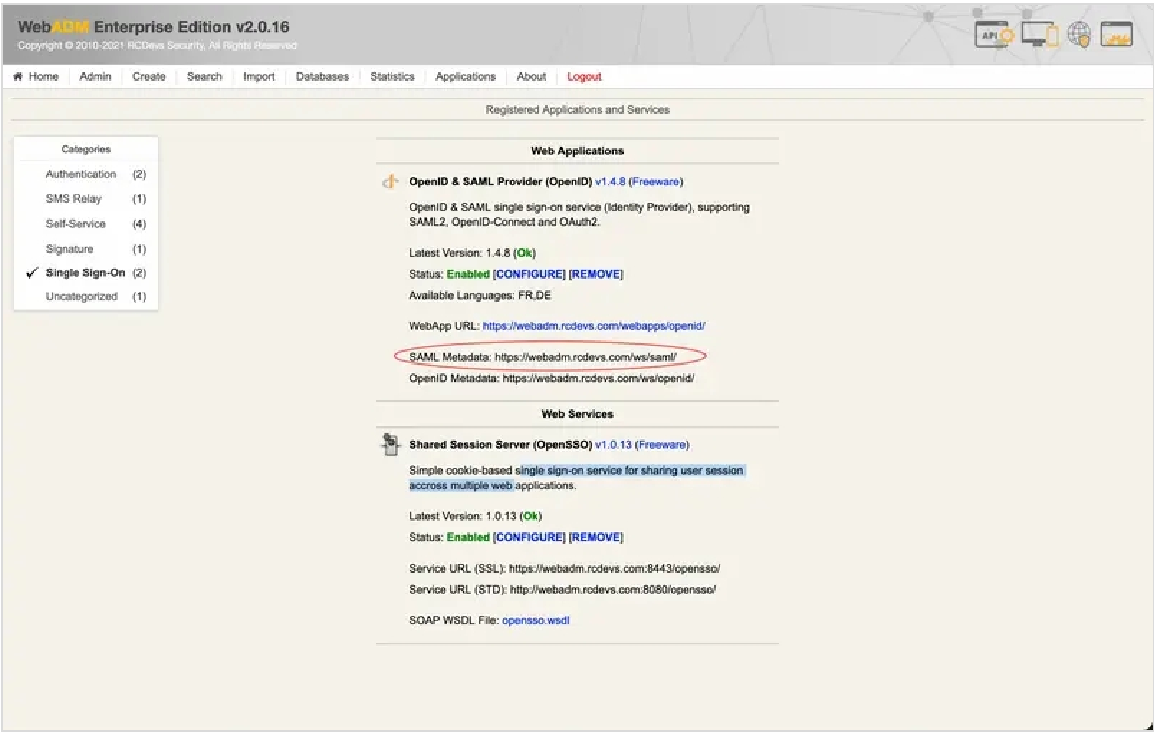
Security defaults are basic identity security mechanisms recommended by Microsoft. When enabled, these recommendations will be automatically enforced in your organization. Administrators and users will be better protected from common identity-related attacks.
Learn more

Enable security defaults
Yes [✓ No]

3.3.6.2 Get your configuration of your IDP on webadm

You will need for the next step

Log on your webadm and go to Applications > Single Sign-On and check the link SAML Metadata



WebADM Enterprise Edition v2.0.16
Copyright © 2010-2021 RCDevs Security, All Rights Reserved
Home
Admin
Create
Search
Import
Databases
Statistics
Applications
About
Logout

Registered Applications and Services

Categories	
Authentication	(2)
SMS Relay	(1)
Self-Service	(4)
Signature	(1)
✓ Single Sign-On	(2)
Uncategorized	(1)

Web Applications

OpenID & SAML Provider (OpenID) v1.4.8 (Freeware)

OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.4.8 (Ok)
Status: Enabled [CONFIGURE] [REMOVE]
Available Languages: FR,DE

WebApp URL: https://webadm.rcdevs.com/webapps/openid/

SAML Metadata: https://webadm.rcdevs.com/ws/saml/
OpenID Metadata: https://webadm.rcdevs.com/ws/openid/

Web Services

Shared Session Server (OpenSSO) v1.0.13 (Freeware)

Simple cookie-based single sign-on service for sharing user session accross multiple web applications.

Latest Version: 1.0.13 (Ok)
Status: Enabled [CONFIGURE] [REMOVE]

Service URL (SSL): https://webadm.rcdevs.com:8443/opensso/
Service URL (STD): http://webadm.rcdevs.com:8080/opensso/

SOAP WSDL File: opensso.wsdl

Figure 3.4.2.1 get your SAML Metadata on WebAdm

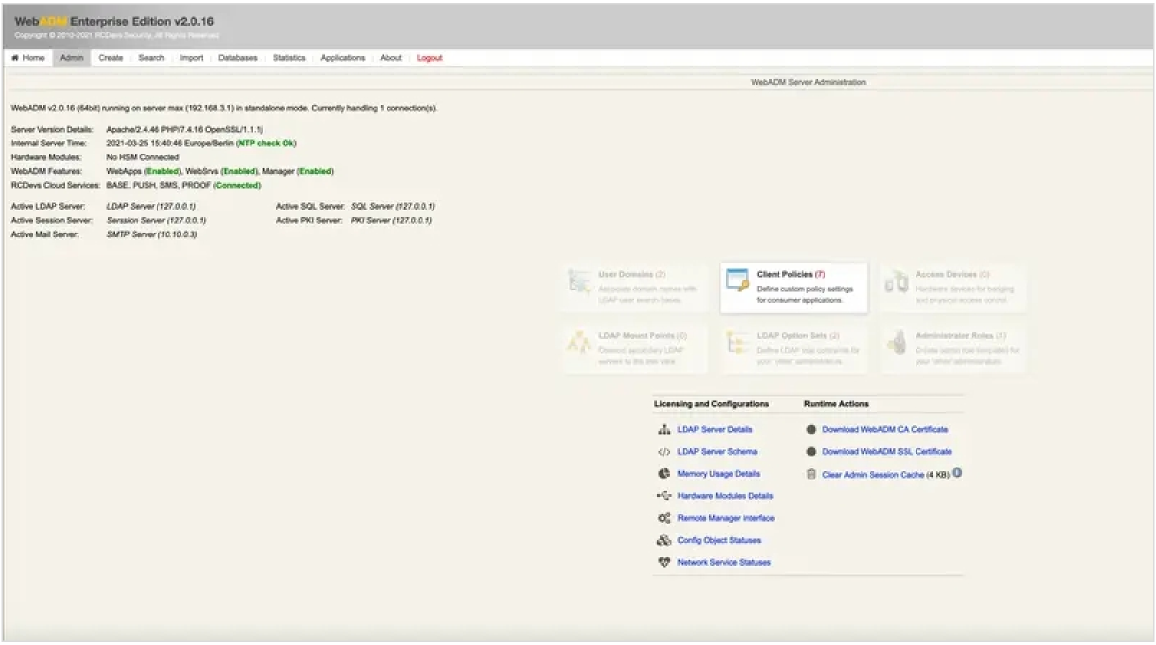
Open the link in a browser

In the XML File you need to get the:

- entityID (https://webadm.foo.bar/)
- X509Certificate (XXXXXXX-X509Certificate-XXXXXXXX)
- SingleSignOnService location (https://webadm.foo.bar/webapps/openid/index.php)

3.3.6.3 Configure propely your IDP and your Policies on webadm

Select Client Policies and go down to click on Add Client



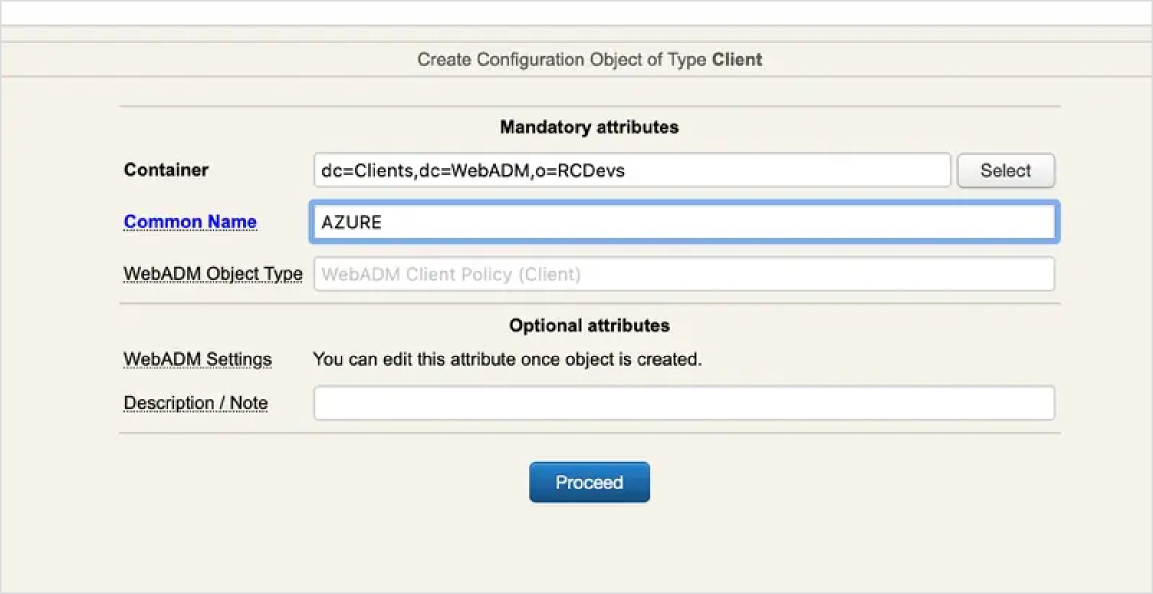
WebADM v2.0.16 (64bit) running on server max (192.168.3.1) in standalone mode. Currently handling 1 connection(s).

Server Version Details: Apache/2.4.46 PHP/7.4.16 OpenSSL/1.1.1j
Internal Server Time: 2021-03-25 15:40:46 Europe/Berlin (NTP check Ok)
Hardware Modules: No HSM Connected
WebADM Features: WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services: BASE, PUSH, SMS, PROOF (Connected)

Active LDAP Server: LDAP Server (127.0.0.1)
Active Session Server: Sersion Server (127.0.0.1)
Active Mail Server: SMTP Server (10.10.0.3)
Active SQL Server: SQL Server (127.0.0.1)
Active PKI Server: PKI Server (127.0.0.1)

Figure 3.4.3.1 Select Client Policy on WebAdm

Give any name in Common Name to your Client Policy (here we use AZURE) Click Proceed then Click on Create Object



Create Configuration Object of Type **Client**

Mandatory attributes

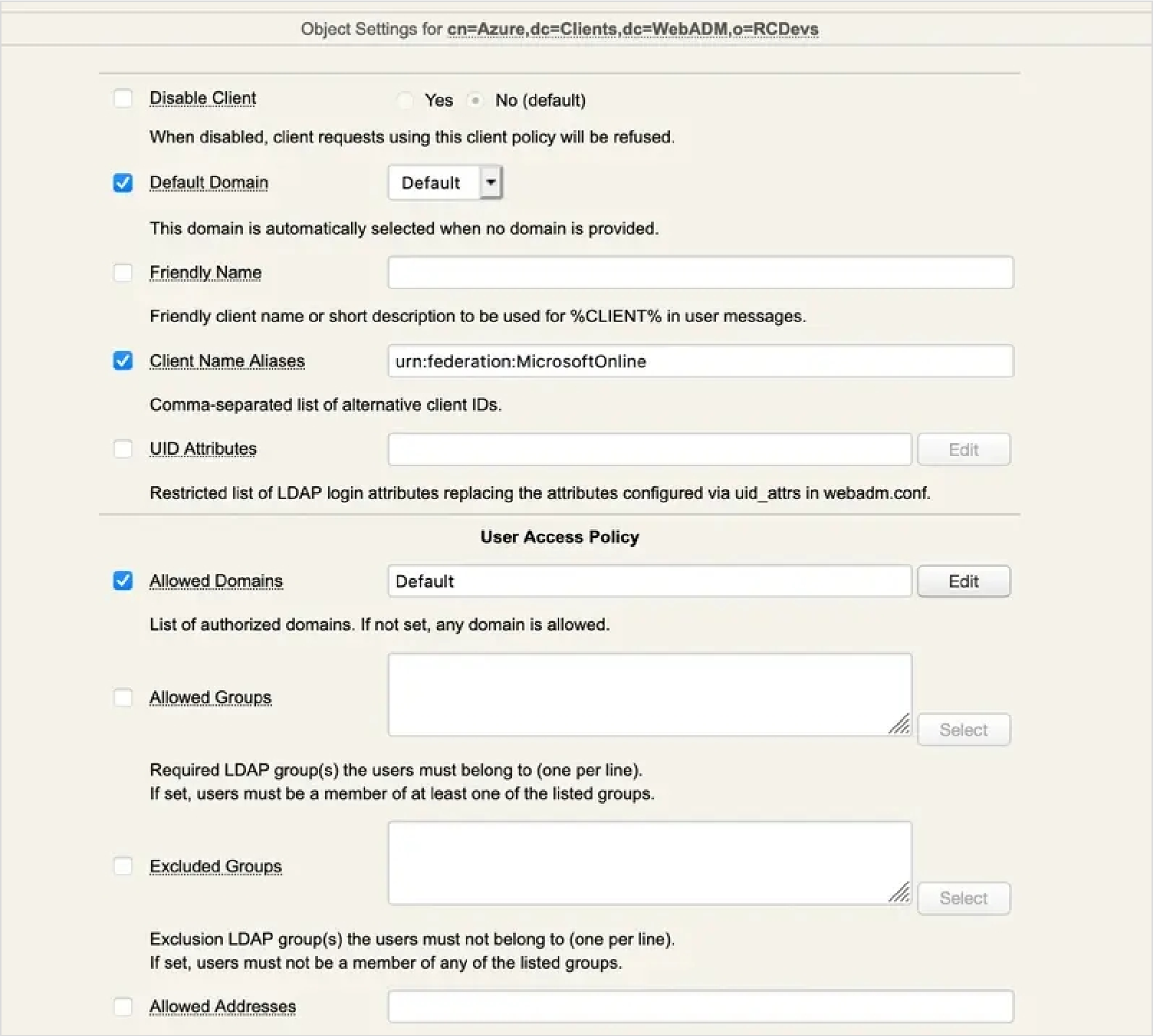
Container	dc=Clients,dc=WebADM,o=RCDevs
Common Name	AZURE
WebADM Object Type	WebADM Client Policy (Client)

Optional attributes

WebADM Settings	You can edit this attribute once object is created.
Description / Note	

Figure 3.4.3.2 Click on Add Client on WebAdm

- Select your Default Domain
- Set your Client Name Aliases to: urn:federation:MicrosoftOnline
- if you have multiple domains set the Allowed Domains to one domain



Object Settings for cn=Azure,dc=Clients,dc=WebADM,o=RCDevs

□ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☑ Default Domain: Default

This domain is automatically selected when no domain is provided.

□ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases: urn:federation:MicrosoftOnline

Comma-separated list of alternative client IDs.

□ UID Attributes

Restricted list of LDAP login attributes replacing the attributes configured via uid_attrs in webadm.conf.

User Access Policy

☑ Allowed Domains: Default

List of authorized domains. If not set, any domain is allowed.

□ Allowed Groups

Required LDAP group(s) the users must belong to (one per line).
If set, users must be a member of at least one of the listed groups.

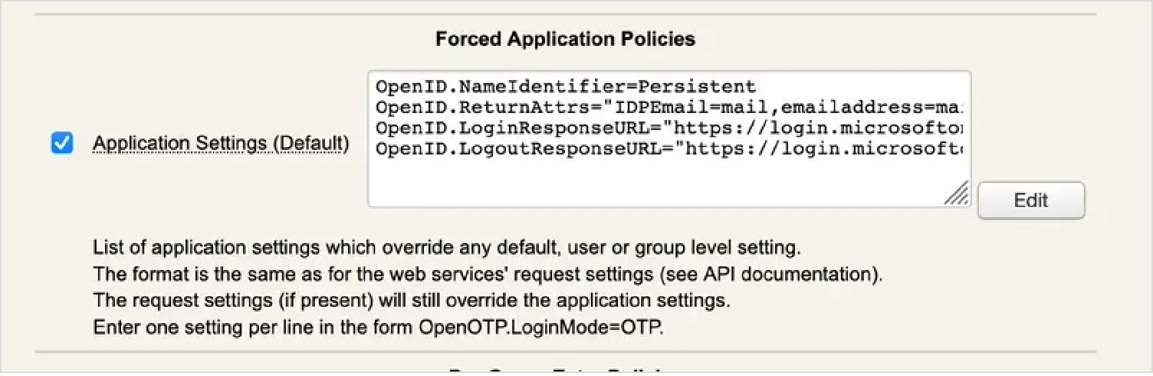
□ Excluded Groups

Exclusion LDAP group(s) the users must not belong to (one per line).
If set, users must not be a member of any of the listed groups.

□ Allowed Addresses

Figure 3.4.3.2 Select your Default Domain in WebAdm

Then click EDIT on Application Settings (Default)



Forced Application Policies

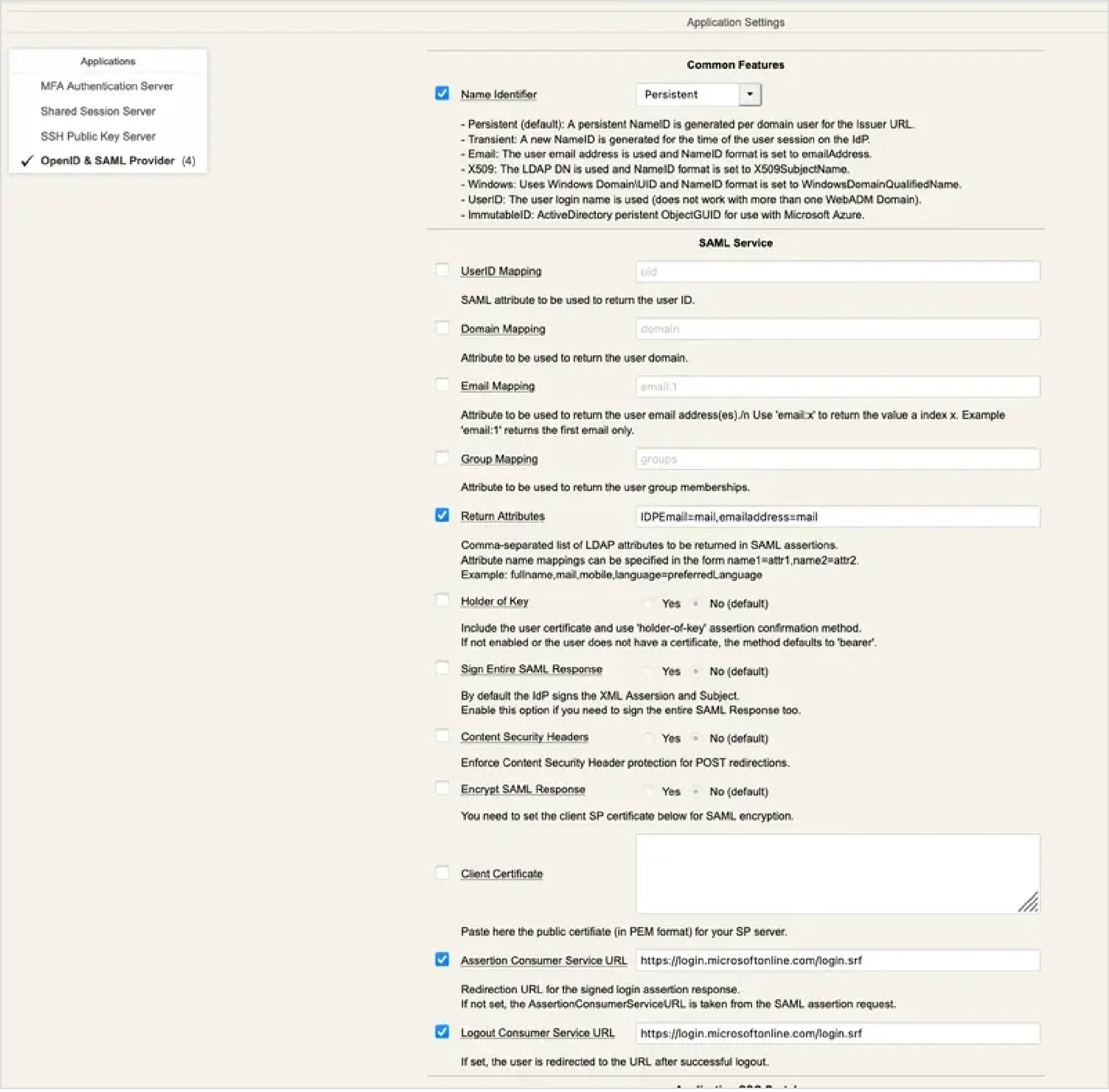
☑ Application Settings (Default)

```
OpenID.NameIdentifier=Persistent
OpenID.ReturnAttrs="IDPEmail=mail,emailaddress=ma
OpenID.LoginResponseURL="https://login.microsofto
OpenID.LogoutResponseURL="https://login.microsoft
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Figure 3.4.3.3 Click EDIT on Application Settings in WebAdm

- Set Name Identifier to Persistent
- Set Return Attributes to IDPEmail=mail,emailaddress=mail with mail our mail attribute in our directoy
- Set Assertion Consumer Service URL to SingleSignOnService location
- Set Logout Consumer Service URL to SingleSignOnService location



Common Features

☑ Name Identifier: Persistent

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

SAML Service

□ UserID Mapping: uid

SAML attribute to be used to return the user ID.

□ Domain Mapping: domain

Attribute to be used to return the user domain.

□ Email Mapping: email:1

Attribute to be used to return the user email address(es)./n Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

□ Group Mapping: groups

Attribute to be used to return the user group memberships.

☑ Return Attributes: IDPEmail=mail,emailaddress=mail

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

□ Holder of Key ○ Yes ● No (default)

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

□ Sign Entire SAML Response ○ Yes ● No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

□ Content Security Headers ○ Yes ● No (default)

Enforce Content Security Header protection for POST redirections.

□ Encrypt SAML Response ○ Yes ● No (default)

You need to set the client SP certificate below for SAML encryption.

□ Client Certificate

Paste here the public certifiate (in PEM format) for your SP server.

☑ Assertion Consumer Service URL: https://login.microsoftonline.com/login.srf

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

☑ Logout Consumer Service URL: https://login.microsoftonline.com/login.srf

If set, the user is redirected to the URL after successful logout.

Figure 3.4.3.3 Set Name Identifier to Persistent in WebAdm

Click on Apply Click Again on Apply It's done !

3.3.6.4 Configure your Domain on AZURE

Lauch a Windows Power Shell

Connect to AZURE with your Administrator

```
PS C:\Users\admin> Connect-MsolService
```

Create your domain (here *foo.bar*)

```
PS C:\Users\admin> New-MsolDomain -Name foo.bar -Authentication Federated
```

You will get in return a CNAME DNS record to add to the dns record of *foo.bar* so Microsoft can verify that you own the domain name. Add the CNAME record to the DNS records of foo.bar. (It could take time to be applied so you could have to wait for the next step)

You will need for the next step

- entityID (https://webadm.foo.bar/)
- X509Certificate (XXXXXXX-X509Certificate-XXXXXXXX)
- SingleSignOnService location (https://webadm.foo.bar/webapps/openid/index.php)

Confirm your domain name

```
PS C:\Users\admin> Confirm-MsolDomain -DomainName foo.bar -IssuerUri https://webadm.foo.bar/ -
FederationBrandName foo.bar -LogOffUri https://webadm.foo.bar/webapps/openid/index.php -
PassiveLogOnUri https://webadm.foo.bar/webapps/openid/index.php -SigningCertificate XXXXXXX-
X509Certificate-XXXXXXXX -PreferredAuthenticationProtocol "SAMLP"
```

Set the Federated authentification methode for your domain

```
PS C:\Users\admin> Set-MSolDomainAuthentication  -DomainName foo.bar -IssuerUri
https://webadm.foo.bar/ -FederationBrandName rcdevs.com -LogOffUri
https://webadm.foo.bar/webapps/openid/index.php -PassiveLogOnUri
https://webadm.foo.bar/webapps/openid/index.php -SigningCertificate XXXXXXX-X509Certificate-
XXXXXXXX -PreferredAuthenticationProtocol "SAMLP" -Authentication Federated
```

3.3.6.5 Get the ImmutableId of your User and add it to Azure

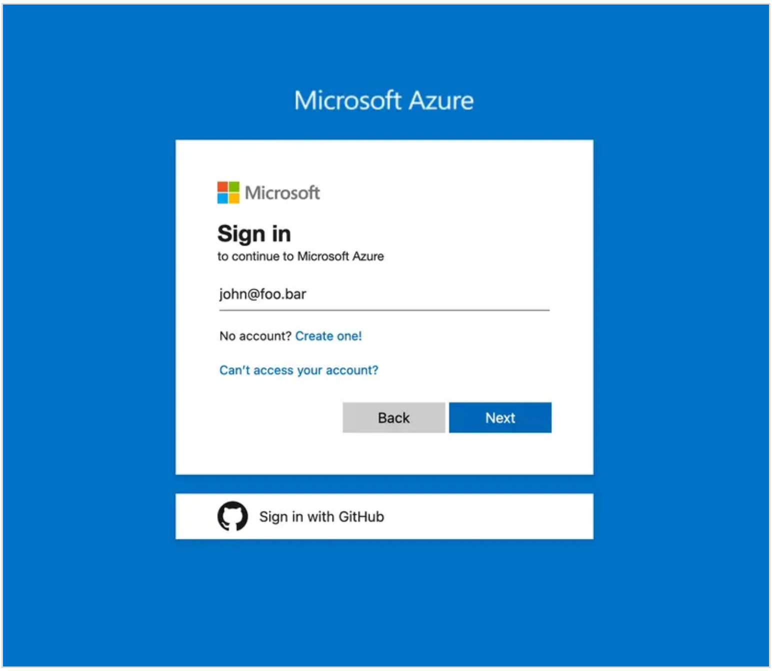
Now you need to add an immutableID for each user in AZURE, but first you need to get this ImmutableId.

(This step is automatic when you use an Active Directory with that is synced with Azure. WebADM/OpenOTP will use your common Object GUID as ImmutableId)

The persistent NameID will be used as ImmutableID. It is generated per domain user for the Issuer URL. It is calculated by the MD5 of the issuer url, followed by /0, followed by the domain, followed by /0 , followed by the username. You can calculate it in a script or use the following method to get it.

Let’s say that you want to log in with the user john@foo.bar

Go on AZURE and initiate a login with the user john@foo.bar.



Microsoft Azure

Sign in

to continue to Microsoft Azure

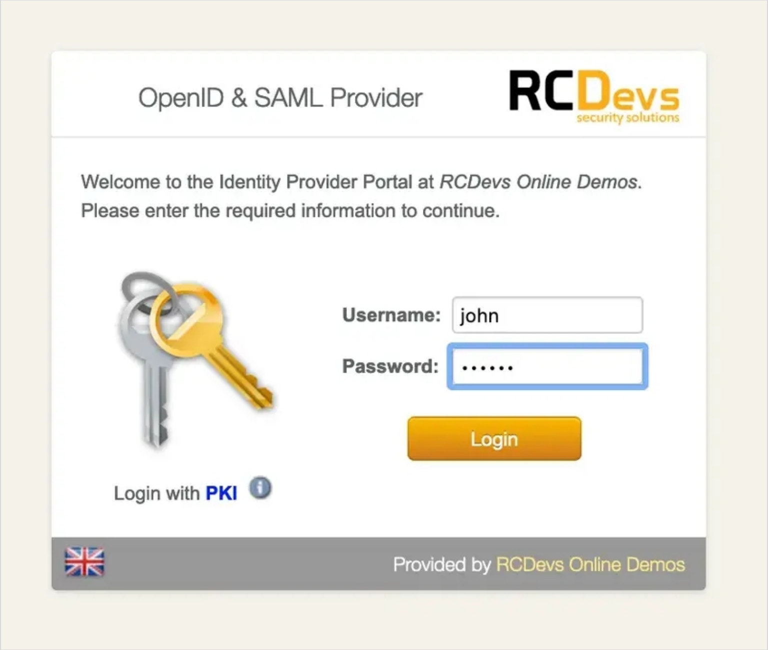
john@foo.bar

No account? Create one!

Can't access your account?

Figure 3.4.5.1 login with the user john@foo.bar on AZURE

It should redirect you on the IDP page to log in



OpenID & SAML Provider

Welcome to the Identity Provider Portal at *RCDevs Online Demos*. Please enter the required information to continue.

Username: john

Password: ••••••

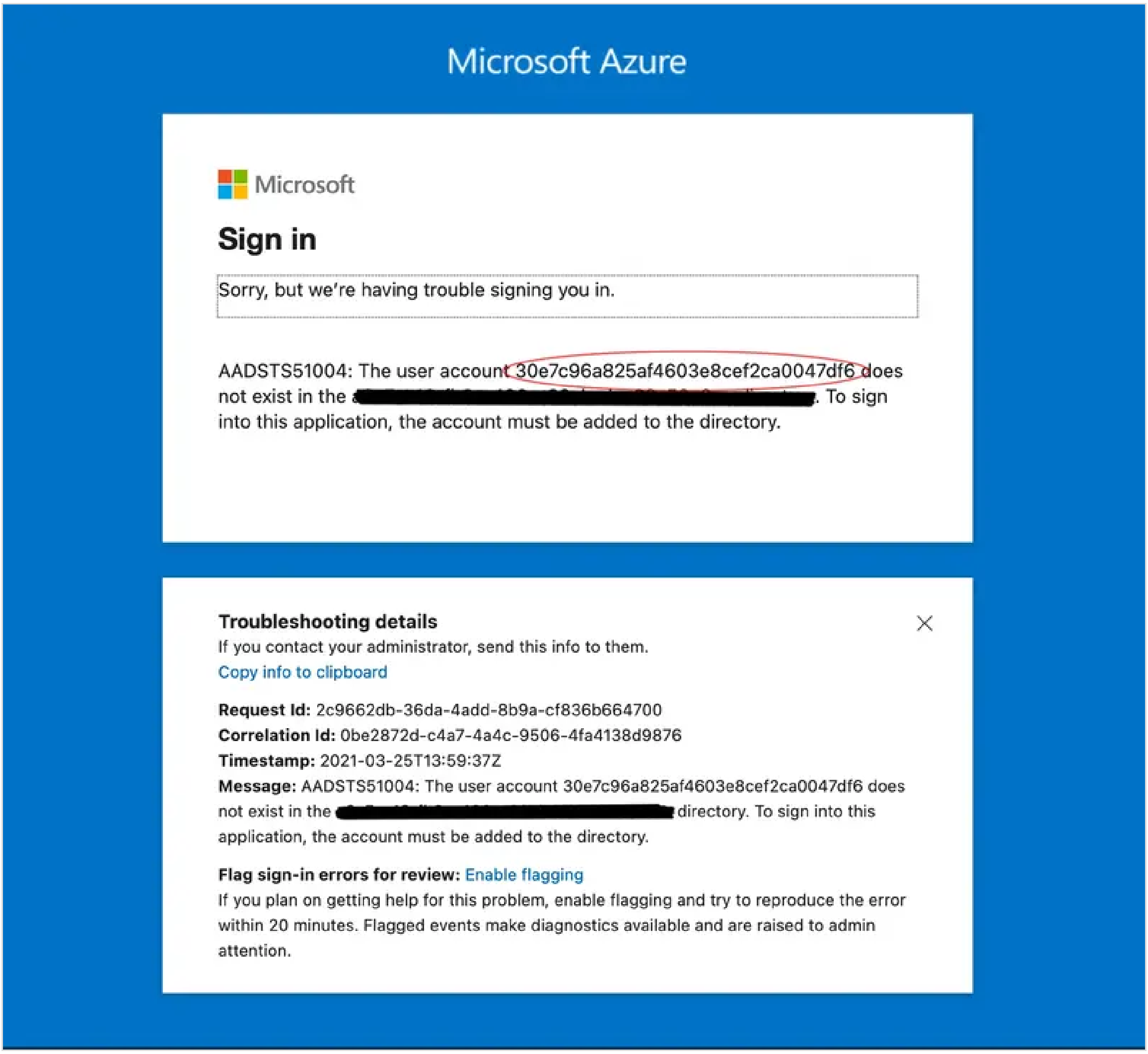
Provided by RCDevs Online Demos

Figure 3.4.5.2 login with the user john@foo.bar on AZURE

Login with your IDP Crediantials

After a succesfull login it will redirect you on the Azure page where it will fail

On the Failed login page you will find your user ImmutableId here 30e7c96a825af4603e8cef2ca0047df6



Sign in

Sorry, but we're having trouble signing you in.

AADSTS51004: The user account 30e7c96a825af4603e8cef2ca0047df6 does not exist in the [redacted]. To sign into this application, the account must be added to the directory.

Troubleshooting details

If you contact your administrator, send this info to them.

Copy info to clipboard

Request Id: 2c9662db-36da-4add-8b9a-cf836b664700
Correlation Id: 0be2872d-c4a7-4a4c-9506-4fa4138d9876
Timestamp: 2021-03-25T13:59:37Z
Message: AADSTS51004: The user account 30e7c96a825af4603e8cef2ca0047df6 does not exist in the [redacted] directory. To sign into this application, the account must be added to the directory.

Flag sign-in errors for review: Enable flagging

If you plan on getting help for this problem, enable flagging and try to reproduce the error within 20 minutes. Flagged events make diagnostics available and are raised to admin attention.

_Figure 3.4.5.3 Failed Login on AZURE where you can find your ImmutableId _

Then you can add your user to AZURE with through PowerShell

```
PS C:\Users\admin> New-MsolUser -UserPrincipalName john@foo.bar -ImmutableId
30e7c96a825af4603e8cef2ca0047df6 -DisplayName "John Doe" -FirstName John -LastName Doe -
AlternateEmailAddresses "john@foo.bar"
```

Now you should be able to log in on the Azure page again. After a successful login on the IDP, you should be redirected and logged into Azure.

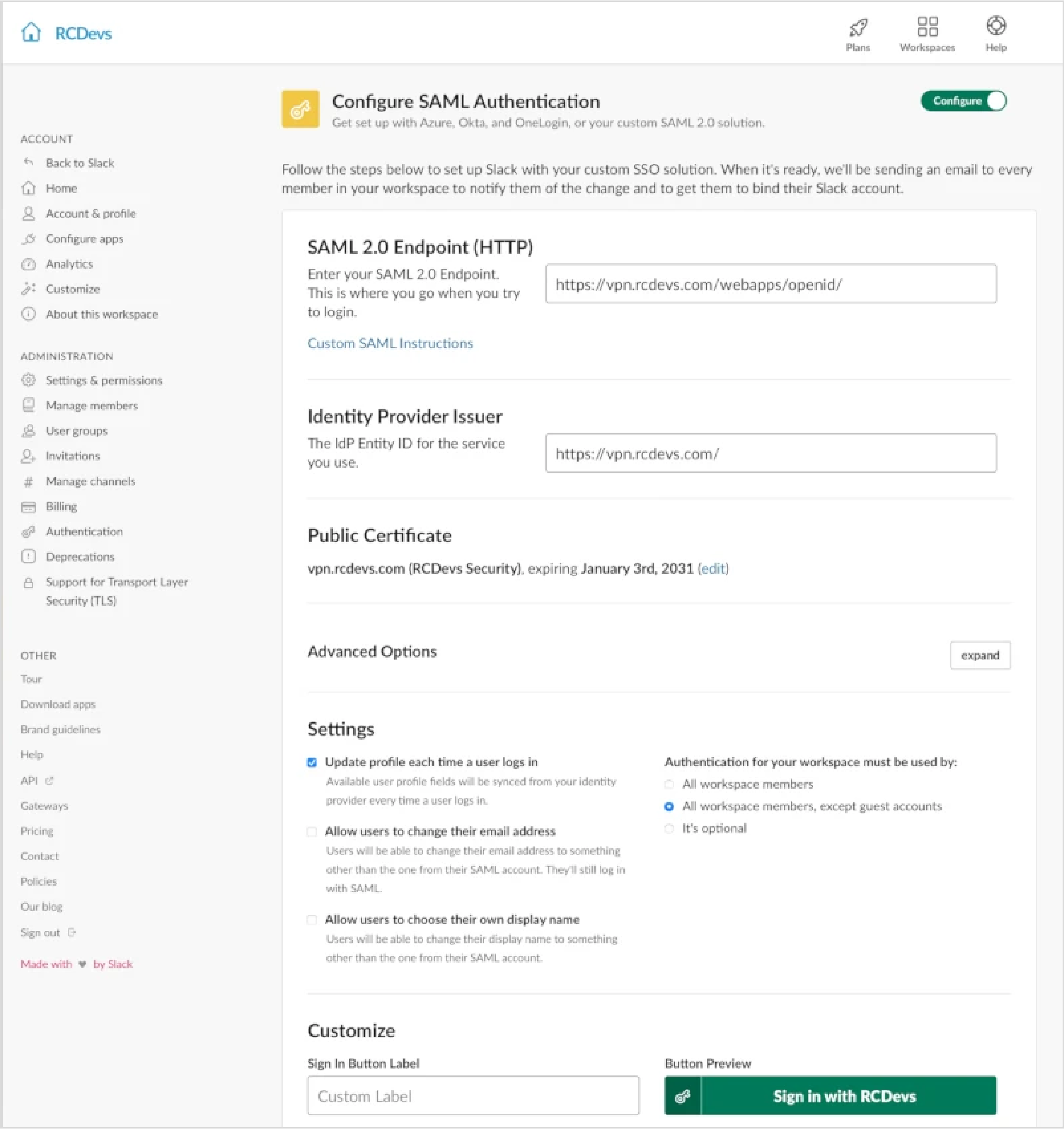
3.3.7 Slack

Have a look on Slack documentation for more information.

3.3.7.1 Slack configuration to use an WebADM IDP (SP configuration)

Login on Slack web page with your Slack administrator account and in Administration category, click on Authentication and configuration your SAML authentication provider. On the SAML configuration page, you have only few settings to configure :

- SAML 2.0 Endpoint
- Identity Provider Issuer



Configure SAML Authentication

Get set up with Azure, Okta, and OneLogin, or your custom SAML 2.0 solution.

Follow the steps below to set up Slack with your custom SSO solution. When it's ready, we'll be sending an email to every member in your workspace to notify them of the change and to get them to bind their Slack account.

SAML 2.0 Endpoint (HTTP)

Enter your SAML 2.0 Endpoint. This is where you go when you try to login.

Custom SAML Instructions

https://vpn.rcdevs.com/webapps/openid/

Identity Provider Issuer

The IdP Entity ID for the service you use.

https://vpn.rcdevs.com/

Public Certificate

vpn.rcdevs.com (RCDevs Security), expiring January 3rd, 2031 (edit)

Advanced Options

Settings

☑ Update profile each time a user logs in
Available user profile fields will be synced from your identity provider every time a user logs in.

□ Allow users to change their email address
Users will be able to change their email address to something other than the one from their SAML account. They'll still log in with SAML.

□ Allow users to choose their own display name
Users will be able to change their display name to something other than the one from their SAML account.

Authentication for your workspace must be used by:

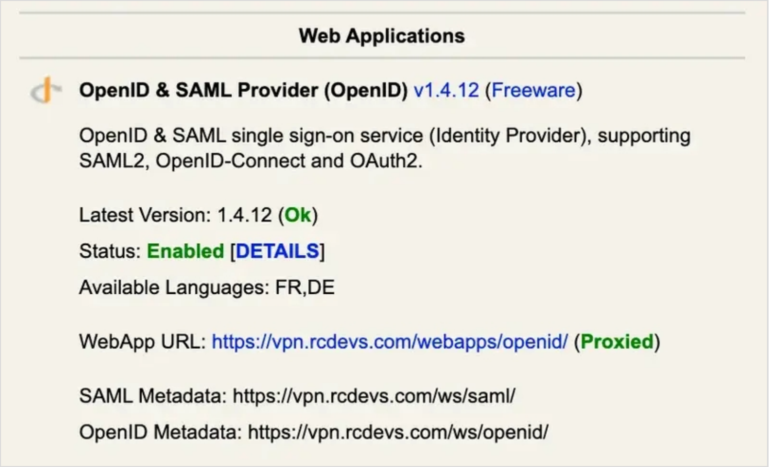
○ All workspace members
● All workspace members, except guest accounts
○ It's optional

Customize

Sign In Button Label: Custom Label

Button Preview

Your SAML 2.0 Endpoint must point to your OpenID application. This information can be found through your `WebADM Admin portal` > `Applications` > `Signle Sign-On` > `WebApp URL`



Web Applications

OpenID & SAML Provider (OpenID) v1.4.12 (Freeware)

OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.4.12 (**Ok**)

Status: **Enabled** [**DETAILS**]

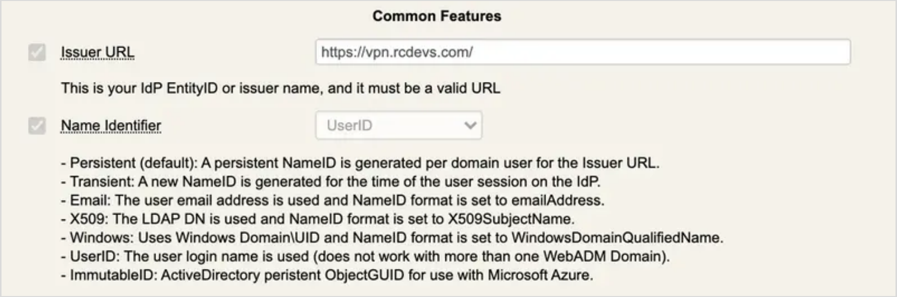
Available Languages: FR,DE

WebApp URL: https://vpn.rcdevs.com/webapps/openid/ (**Proxied**)

SAML Metadata: https://vpn.rcdevs.com/ws/saml/

OpenID Metadata: https://vpn.rcdevs.com/ws/openid/

The identity provider issuer (Issuer URL) can be found under the OpenID & SAML Provider configuration.



Common Features

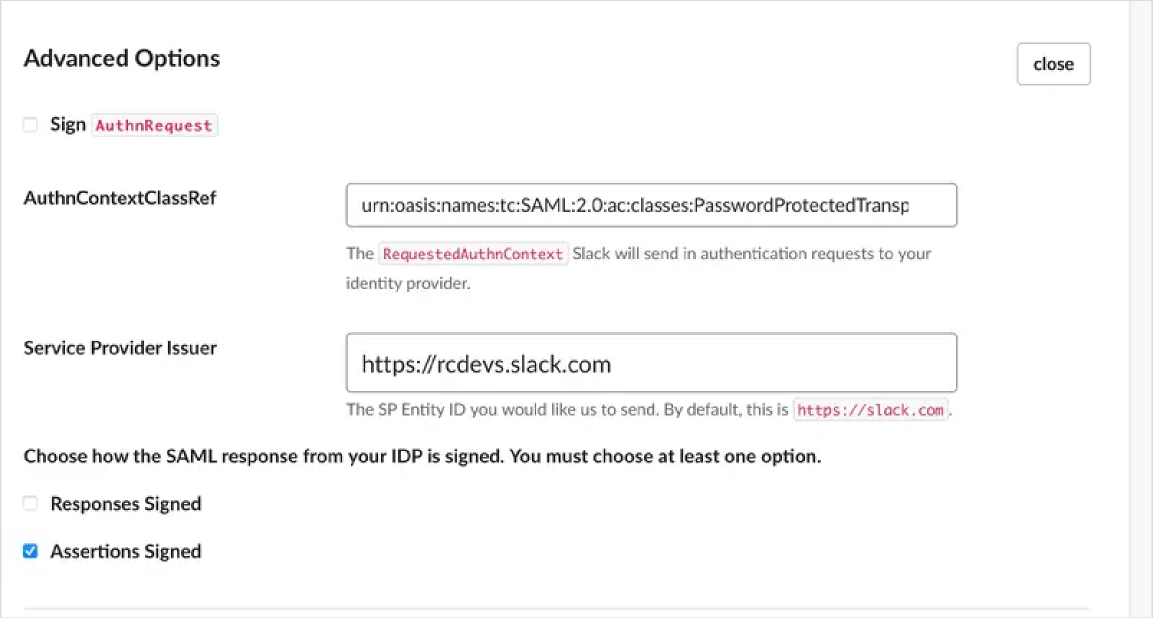
☑ Issuer URL: https://vpn.rcdevs.com/

This is your IdP EntityID or issuer name, and it must be a valid URL

☑ Name Identifier: UserID

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

In advanced options on Slack, you musst configure the following :



Advanced Options

□ Sign `AuthnRequest`

AuthnContextClassRef: urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransp

The `RequestedAuthnContext` Slack will send in authentication requests to your identity provider.

Service Provider Issuer: https://rcdevs.slack.com

The SP Entity ID you would like us to send. By default, this is `https://slack.com`.

Choose how the SAML response from your IDP is signed. You must choose at least one option.

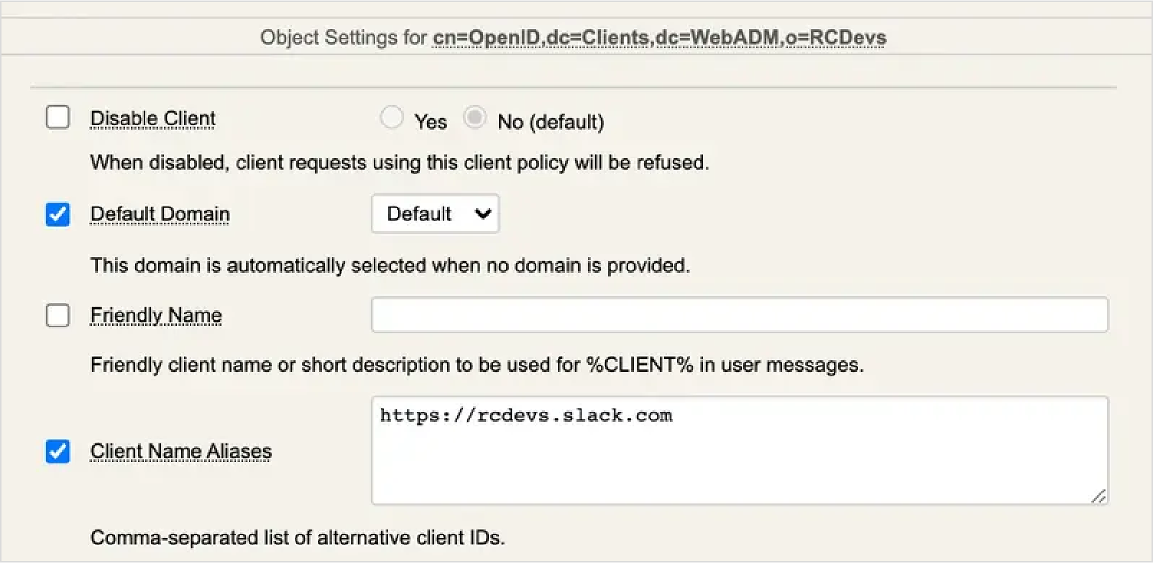
□ Responses Signed

☑ Assertions Signed

The Service Provider Issuer should point to https://slack.com or https://your_slack_domain.slack.com, this setting will be used later to match a WebADM client policy. You must enable the setting Assertions Signed.

3.3.7.2 Configure a WebADM client policy for Slack

You can now create a client policy for Slack and apply specific SAML/OpenID or OpenOTP settings inside that policy. In `client name aliases` setting of your `WebADM client policy`, you must configure the value you configure as Service Provider Issuer on Slack admin console.



Object Settings for cn=OpenID,dc=Clients,dc=WebADM,o=RCDevs

□ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☑ Default Domain Default

This domain is automatically selected when no domain is provided.

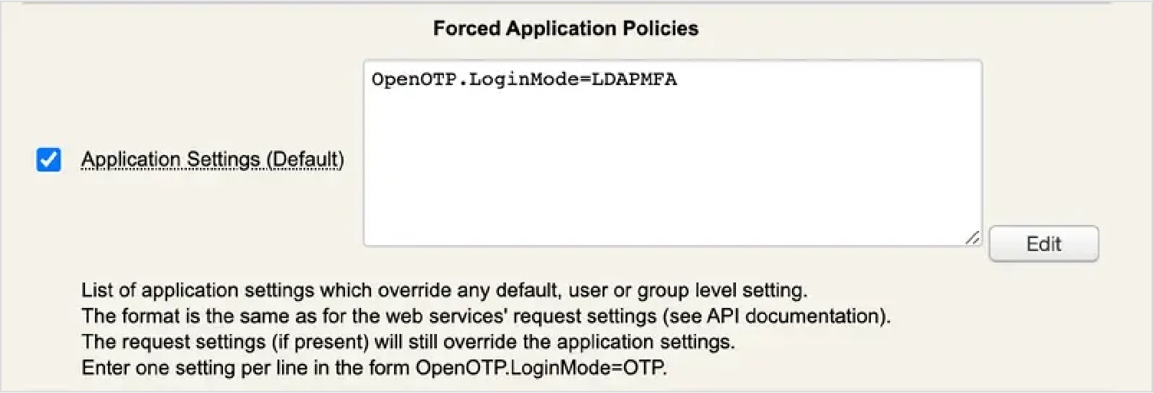
□ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases `https://rcdevs.slack.com`

Comma-separated list of alternative client IDs.

And you configure OpenOTP setting as below :



Forced Application Policies

☑ Application Settings (Default) `OpenOTP.LoginMode=LDAPMFA`

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

3.3.7.3 Authentication logs for Slack

```
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] Enforcing client policy: OpenID
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] New login request (OpenOTP)
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] > Client ID: OpenID
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] > Username: support
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] > Domain: Default
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] > ANY Password: xxxxxxxxxxxxxx
[2021-07-22 07:07:41] [192.168.3.254:50416] [OpenID:OTDHTF8T] Sending openotpSimpleLogin request

[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] New openotpSimpleLogin SOAP
request
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Username: support
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Domain: Default
```

```
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Domain: Default
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Password: xxxxxxxxxxxxxx
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Client ID: OpenID
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] > Source IP: 87.123.192.156
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Enforcing client policy: OpenID
(matched client ID)
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Registered openotpSimpleLogin
request
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Resolved LDAP user:
uid=support,ou=Users,o=RCDevs (cached)
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Resolved LDAP groups: staff,support
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Resolved source location: DE
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Started transaction lock for user
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Found user fullname: support
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Found 2 user
emails:support@rcdevs.com
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Found 48 user settings:
LoginMode=LDAPMFA,OTPType=TOKEN,PushLogin=Yes,PushVoice=Yes,ChallengeMode=Yes,ChallengeTime
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=FIDO2,U2FPINMode=Discouraged,SMSType=Normal,SMSMode=Ondemand,MailMode=Onc

[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Found 6 user data:
AppKeyInit,TokenType,TokenKey,TokenState,TokenID,TokenSerial
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Found 1 registered OTP token (TOTP)
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] User has no FIDO device registered
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Requested login factors: LDAP & OTP
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] LDAP password Ok
[2021-07-22 07:07:41] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Authentication challenge required
[2021-07-22 07:07:42] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Sent push notification for token #1
(session z5ilnF3a6d3Iwz06)
[2021-07-22 07:07:42] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Waiting 27 seconds for mobile
response
[2021-07-22 07:07:53] [192.168.3.254:50422] [OpenOTP:OTDHTF8T] Received mobile login response
from 194.31.54.217
[2021-07-22 07:07:53] [192.168.3.254:50422] [OpenOTP:OTDHTF8T] > Session: z5ilnF3a6d3Iwz06
[2021-07-22 07:07:53] [192.168.3.254:50422] [OpenOTP:OTDHTF8T] > Password: 16 Bytes
[2021-07-22 07:07:53] [192.168.3.254:50422] [OpenOTP:OTDHTF8T] Found authentication session
started 2021-07-22 07:07:41
[2021-07-22 07:07:53] [192.168.3.254:50422] [OpenOTP:OTDHTF8T] PUSH password Ok (token #1)
[2021-07-22 07:07:53] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Updated user data
[2021-07-22 07:07:53] [192.168.3.1:59726] [OpenOTP:OTDHTF8T] Sent login success response

[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] OpenOTP authentication success
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Resolved LDAP user:
uid=support,ou=Users,o=RCDevs (cached)
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Resolved LDAP groups: staff,support
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Resolved source location: DE
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Login session started for
uid=support,ou=Users,o=RCDevs
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Returning nameId value 'support'
```

```
[2021-07-22 07:07:53] [192.168.3.254:50416] [OpenID:OTDHTF8T] Sent SAML login success response
```

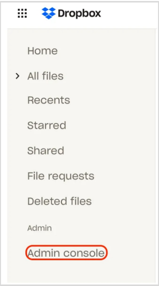
3.3.8 Dropbox

Tested on Dropbox Business.

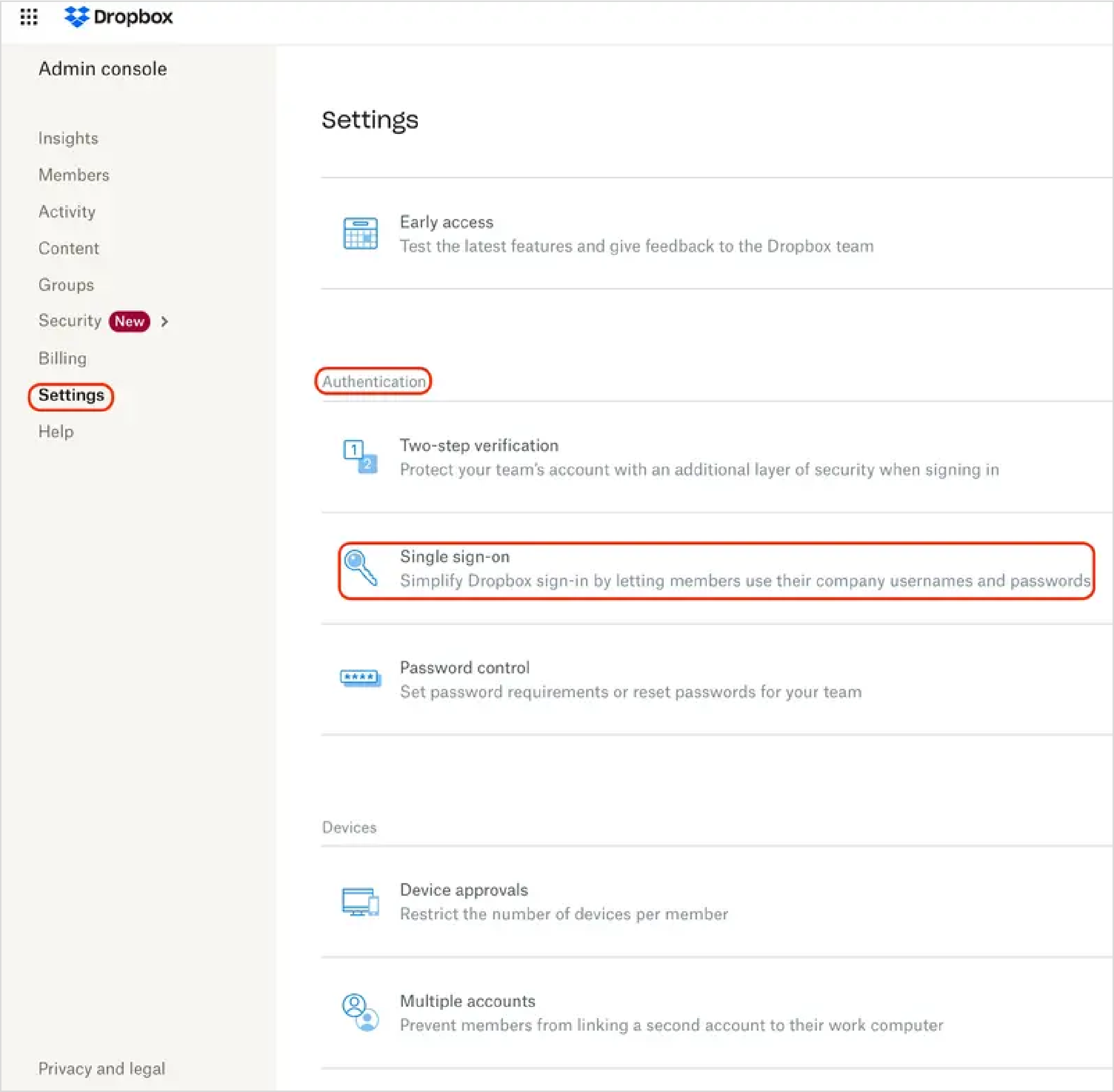
Note

Firstly for **Dropbox** side, each user should have their own account. Join the Business team normally and get a license. Then in Webadm this user must have their Dropbox Email in the attribute : Email Address.
For example if I am subscribed to **Dropbox** with this email address: example@mail.com, I must have this email added in **Email Address attribute** in Webadm as well.

After sign in to **Dropbox** using your admin credentials, Select **Admin console :**



Navigate to **Settings > Authentication > Single sign-on :**



Settings

Early access
Test the latest features and give feedback to the Dropbox team

Authentication

Two-step verification
Protect your team's account with an additional layer of security when signing in

Single sign-on
Simplify Dropbox sign-in by letting members use their company usernames and passwords

Password control
Set password requirements or reset passwords for your team

Devices

Device approvals
Restrict the number of devices per member

Multiple accounts
Prevent members from linking a second account to their work computer

Enter the following information :

1- Single sign-on : Select the appropriate option

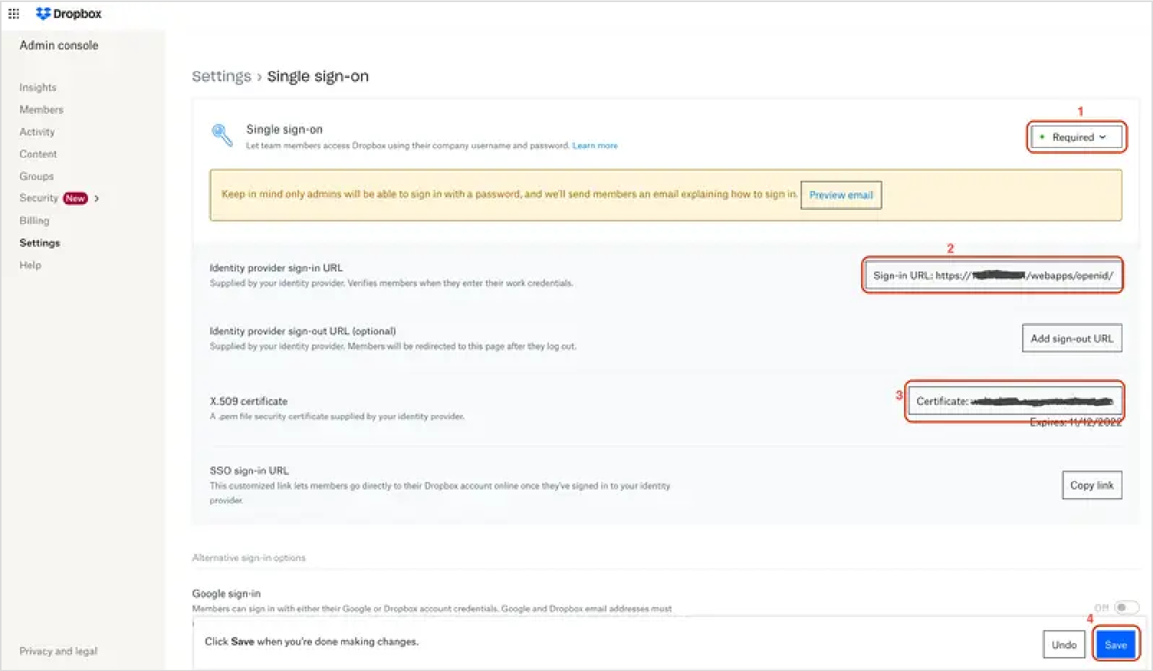
2- Identity provider sign-in URL:

This information can be found through your **WebADM Admin portal > Applications > Single Sign-On > WebApp URL**

3- X.509 certificate : Upload the following: (PEM format)

WebADM Admin portal > Applications > Signle Sign-On > [CONFIGURE] > Common Features > Server Certificate.

4- Click **Save**.



Settings › Single sign-on

Single sign-on
Let team members access Dropbox using their company username and password. Learn more

• Required

Keep in mind only admins will be able to sign in with a password, and we'll send members an email explaining how to sign in. Preview email

Identity provider sign-in URL
Supplied by your identity provider. Verifies members when they enter their work credentials.

Sign-in URL: https://[redacted]/webapps/openid/

Identity provider sign-out URL (optional)
Supplied by your identity provider. Members will be redirected to this page after they log out.

Add sign-out URL

X.509 certificate
A .pem file security certificate supplied by your identity provider.

Certificate: [redacted]
Expires: 11/12/2022

SSO sign-in URL
This customized link lets members go directly to their Dropbox account online once they've signed in to your identity provider.

Copy link

Alternative sign-in options

Google sign-in
Members can sign in with either their Google or Dropbox account credentials. Google and Dropbox email addresses must

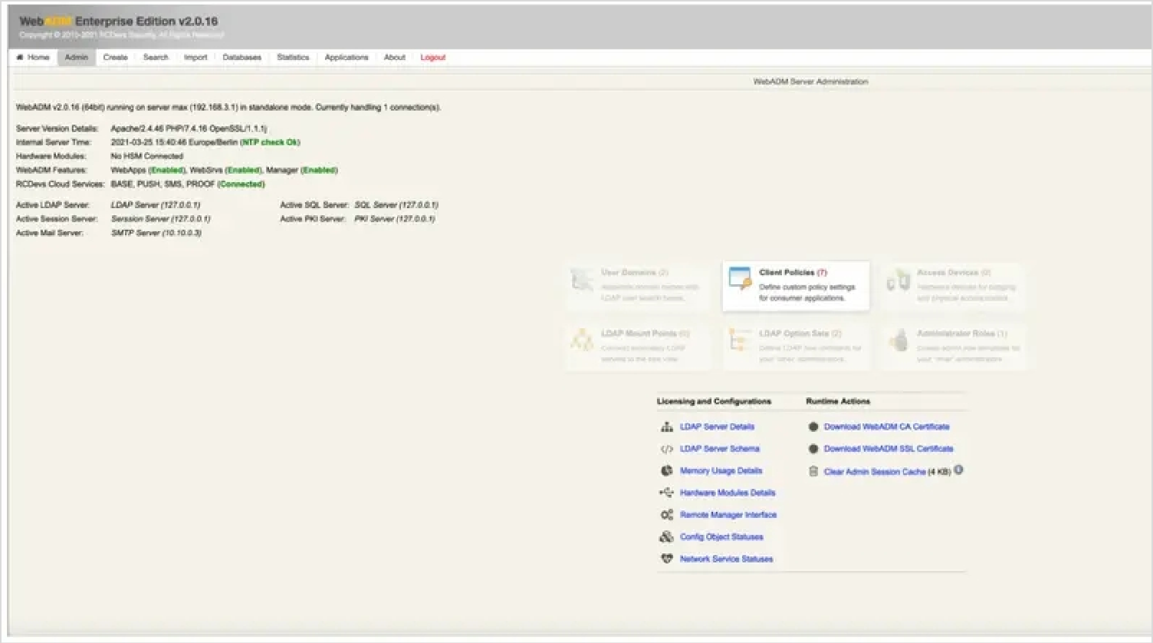
Off

Click **Save** when you're done making changes.

Undo Save

Configure propely your IDP and your Policy on webadm

- Select Client Policy and go down to click on Add Client :

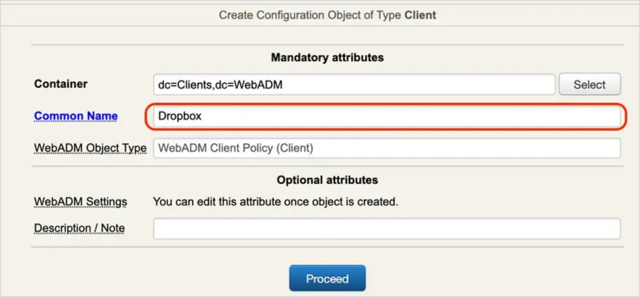


WebADM v2.0.16 (64bit) running on server max (192.168.3.1) in standalone mode. Currently handling 1 connection(s).

Server Version Details:	Apache/2.4.46 PHP/7.4.16 OpenSSL/1.1.1j
Internal Server Time:	2021-03-25 15:40:46 Europe/Berlin (NTP check Ok)
Hardware Modules:	No HSM Connected
WebADM Features:	WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services:	BASE, PUSH, SMS, PROOF (Connected)

Active LDAP Server:	LDAP Server (127.0.0.1)	Active SQL Server:	SQL Server (127.0.0.1)
Active Session Server:	Session Server (127.0.0.1)	Active PKI Server:	PKI Server (127.0.0.1)
Active Mail Server:	SMTP Server (10.10.0.3)		

Give any name in Common Name to your **Client Policy** (here we use Dropbox), Click **Proceed** then Click on **Create Object** :



Create Configuration Object of Type **Client**

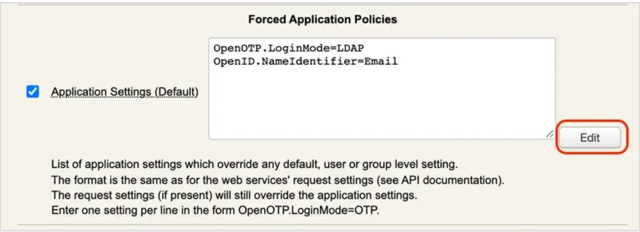
Mandatory attributes

Container	dc=Clients,dc=WebADM
Common Name	Dropbox
WebADM Object Type	WebADM Client Policy (Client)

Optional attributes

WebADM Settings	You can edit this attribute once object is created.
Description / Note	

Then click **EDIT on Application Settings (Default) :**



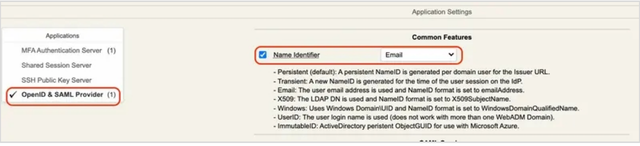
Forced Application Policies

☑ Application Settings (Default)

```
OpenOTP.LoginMode=LDAP
OpenID.NameIdentifier=Email
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Set **Name Identifier** to **Email** :



Application Settings

Common Features

☑ Name Identifier: Email

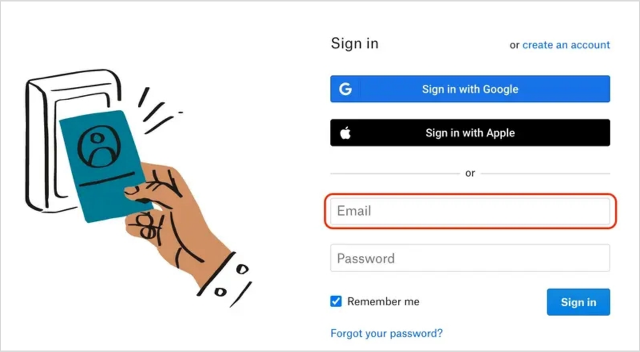
- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

Your **Dropbox** user must also be created in webADM with **Email address attribute.**

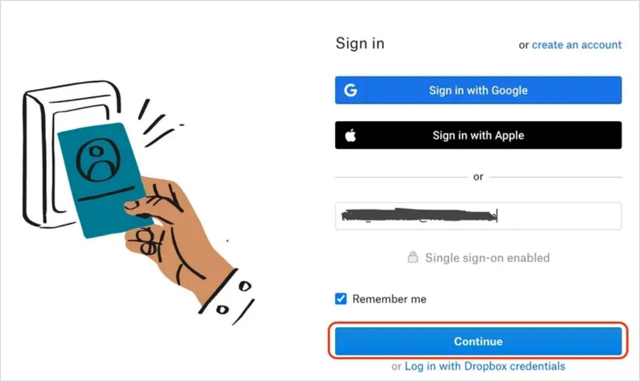
> ### SSO Authentication :

Go to https://www.dropbox.com/login.

Enter your Email:



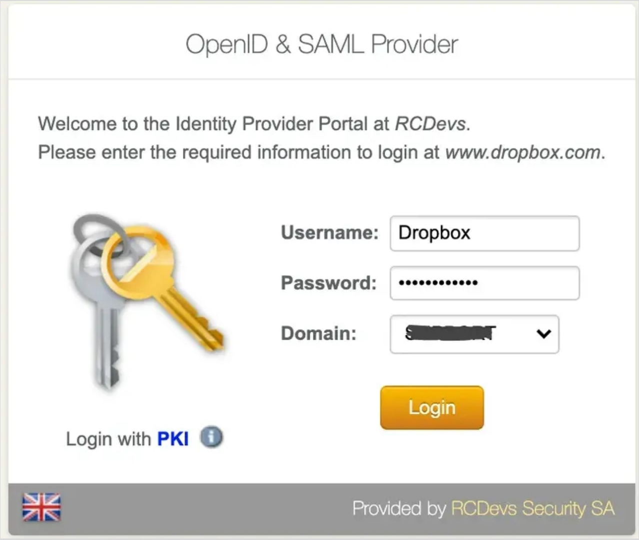
Click Continue:



[redacted]

Single sign-on enabled

Login with your user created in WebADM/Dropbox :



OpenID & SAML Provider

Welcome to the Identity Provider Portal at *RCDevs*.
Please enter the required information to login at *www.dropbox.com*.

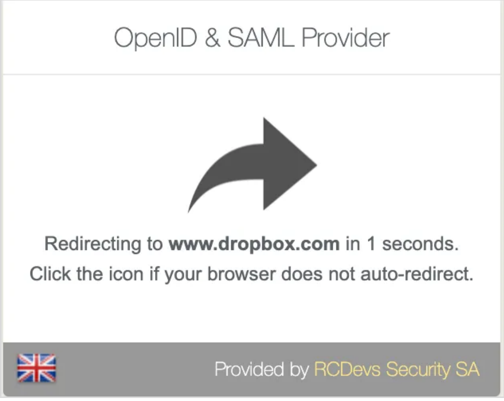
Username: Dropbox
Password: ••••••••••••
Domain: [redacted]

Login

Login with **PKI**

Provided by RCDevs Security SA

After **Successful Authentication** you are redirected to the Dropbox SP :



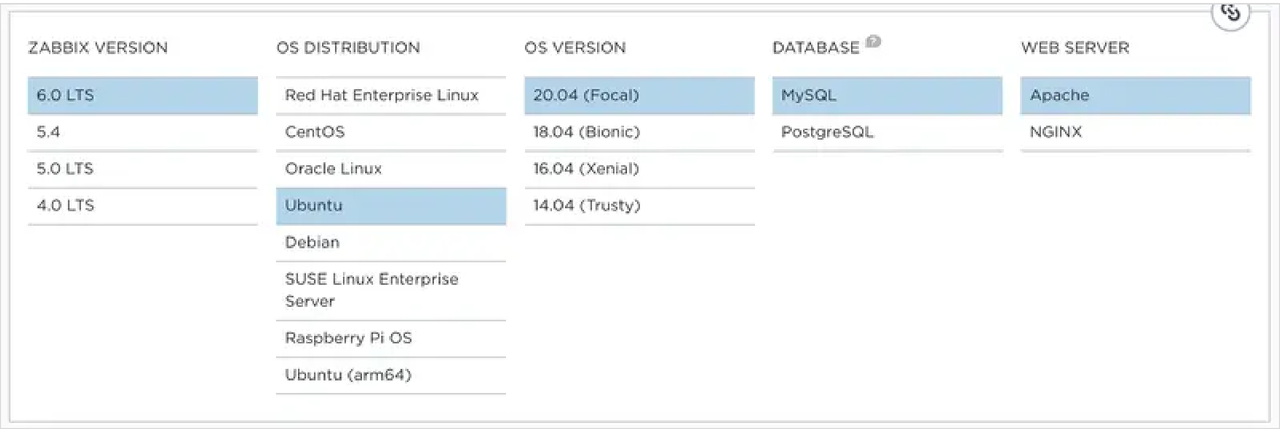
OpenID & SAML Provider

Redirecting to **www.dropbox.com** in 1 seconds.
Click the icon if your browser does not auto-redirect.

Provided by RCDevs Security SA

3.3.9 Zabbix

Tested with the following configuration :



ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	DATABASE	WEB SERVER
6.0 LTS	Red Hat Enterprise Linux	20.04 (Focal)	MySQL	Apache
5.4	CentOS	18.04 (Bionic)	PostgreSQL	NGINX
5.0 LTS	Oracle Linux	16.04 (Xenial)		
4.0 LTS	Ubuntu	14.04 (Trusty)		
	Debian			
	SUSE Linux Enterprise Server			
	Raspberry Pi OS			
	Ubuntu (arm64)			

We will start by adding a Public Certificate to **Zabbix :**

In your server uncomment this line :

```
vi /etc/zabbix/web/zabbix.conf.php
$SSO['IDP_CERT']        = 'conf/certs/idp.crt';
```

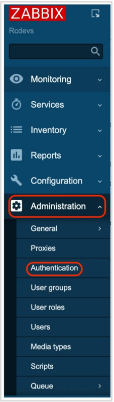
Create a new file **idp.crt** in this path : **/usr/share/zabbix/conf/certs** and put inside the public Certificate which is in : WebADM > Application > Single Sign-on > Public Certifiate.

Restart **Zabbix server** and **agent processes :**

```
systemctl restart zabbix-server zabbix-agent apache2
```

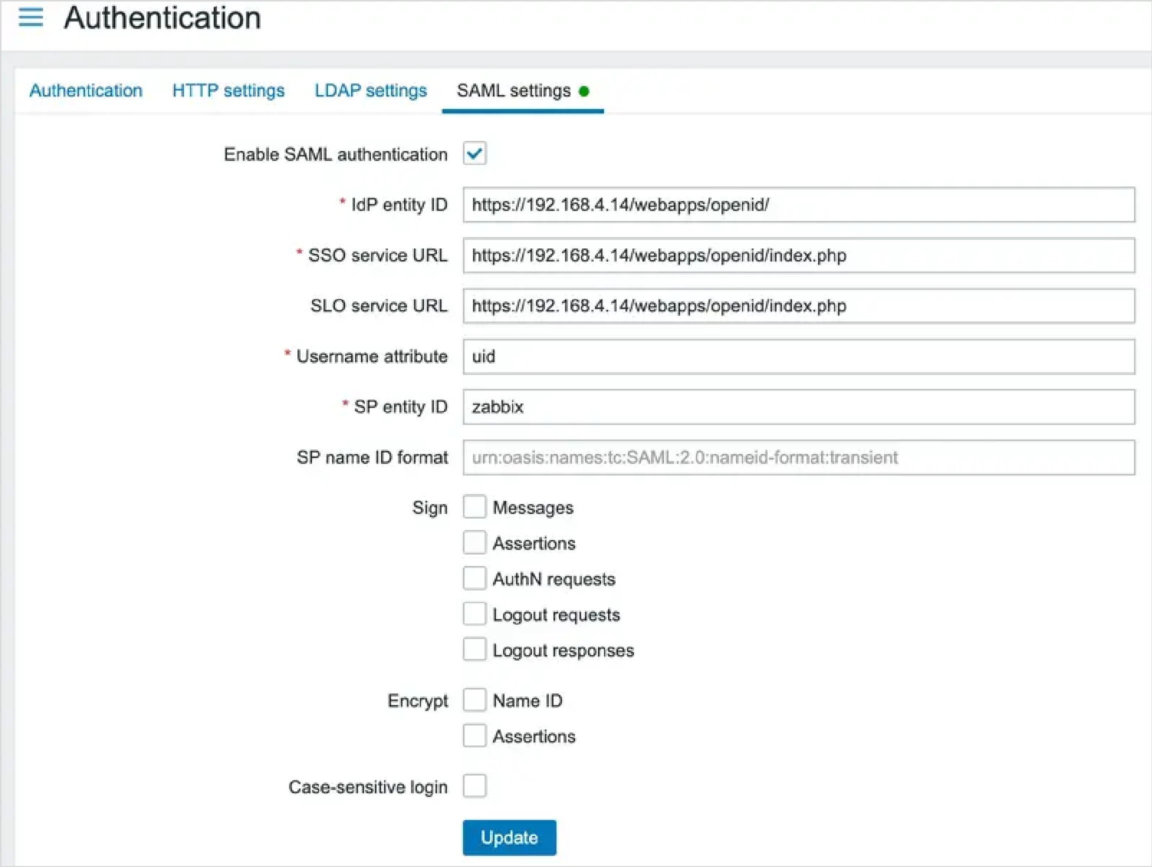
After sign in to Zabbix web interface, Navigate to **Administration** > **Authentication**.

Note that a user must exist in Zabbix. If authentication is successful, then Zabbix will match a local username with the username attribute returned by SAML.



Select the **SAML settings** tab and Enable **SAML authentication** check box then Enter the following information:

- **IdP entity ID**, **SSO service URL**, **SLO service URL**: Values from **WebADM** > **Applications** > **Single Sign-On** > **OpenID & SAML Provider.**
- **Username attribute**: uid.
- **SP entity ID**: zabbix (You specify this value when you configure a client Policy in the next step).
- **Click Update.**



Enable SAML authentication ☑

* IdP entity ID: https://192.168.4.14/webapps/openid/

* SSO service URL: https://192.168.4.14/webapps/openid/index.php

SLO service URL: https://192.168.4.14/webapps/openid/index.php

* Username attribute: uid

* SP entity ID: zabbix

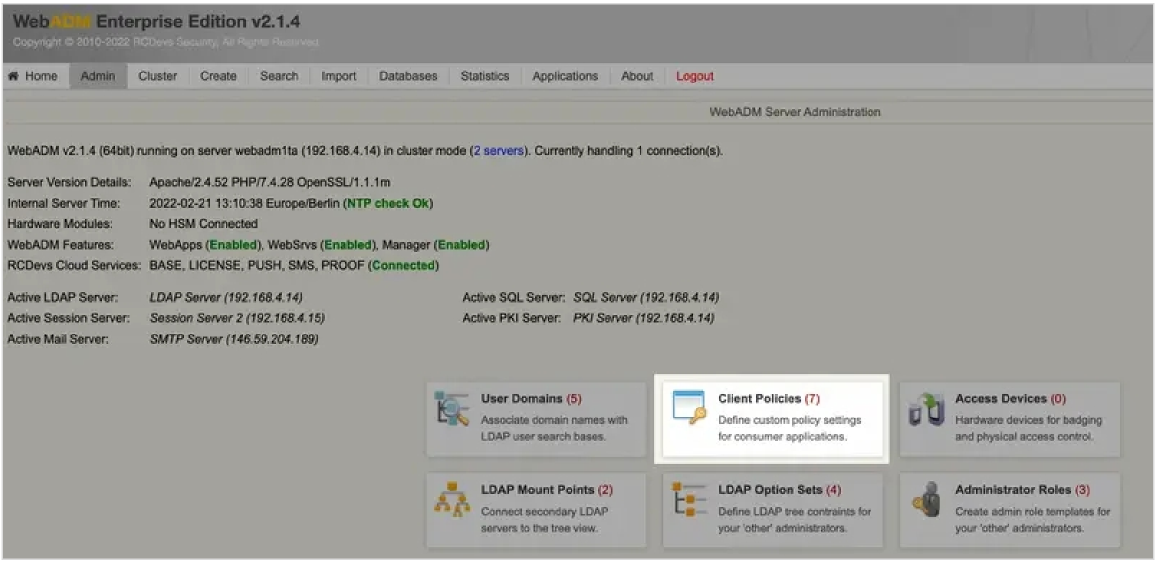
SP name ID format: urn:oasis:names:tc:SAML:2.0:nameid-format:transient

Sign: □ Messages □ Assertions □ AuthN requests □ Logout requests □ Logout responses

Encrypt: □ Name ID □ Assertions

Case-sensitive login □

Configure your Policy on webadm:


WebADM Enterprise Edition v2.1.4
Copyright © 2010-2022 RCDevs Security, All Rights Reserved
Home
Admin
Cluster
Create
Search
Import
Databases
Statistics
Applications
About
Logout
WebADM Server Administration

WebADM v2.1.4 (64bit) running on server webadm1ta (192.168.4.14) in cluster mode (2 servers). Currently handling 1 connection(s).

Server Version Details: Apache/2.4.52 PHP/7.4.28 OpenSSL/1.1.1m
Internal Server Time: 2022-02-21 13:10:38 Europe/Berlin (NTP check Ok)
Hardware Modules: No HSM Connected
WebADM Features: WebApps (Enabled), WebSrvs (Enabled), Manager (Enabled)
RCDevs Cloud Services: BASE, LICENSE, PUSH, SMS, PROOF (Connected)

Active LDAP Server: *LDAP Server (192.168.4.14)*
Active Session Server: *Session Server 2 (192.168.4.15)*
Active Mail Server: *SMTP Server (146.59.204.189)*
Active SQL Server: *SQL Server (192.168.4.14)*
Active PKI Server: *PKI Server (192.168.4.14)*

User Domains (5) Associate domain names with LDAP user search bases.

Client Policies (7) Define custom policy settings for consumer applications.

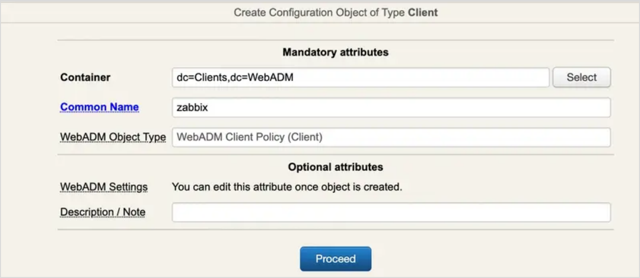
Access Devices (0) Hardware devices for badging and physical access control.

LDAP Mount Points (2) Connect secondary LDAP servers to the tree view.

LDAP Option Sets (4) Define LDAP tree contraints for your 'other' administrators.

Administrator Roles (3) Create admin role templates for your 'other' administrators.

Give any name in Common Name to your **Client Policy** (here we use Zabbix), Click **Proceed** then Click on **Create Object** :



Create Configuration Object of Type **Client**

Mandatory attributes

Container: dc=Clients,dc=WebADM

Common Name: zabbix

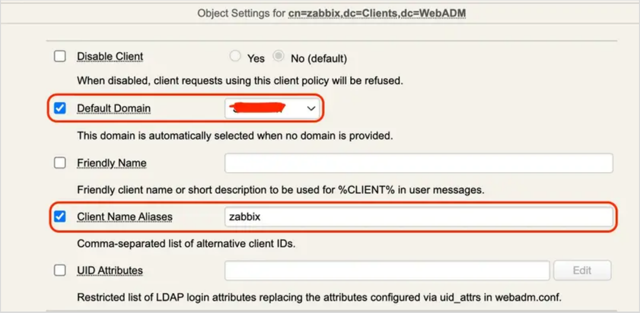
WebADM Object Type: WebADM Client Policy (Client)

Optional attributes

WebADM Settings: You can edit this attribute once object is created.

Description / Note:

Set here your **Domain**, and enter the **Client Name Aliases** that you configured before in **SP entity ID** (Zabbix side)



Object Settings for cn=zabbix,dc=Clients,dc=WebADM

☐ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☑ Default Domain [redacted]

This domain is automatically selected when no domain is provided.

☐ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

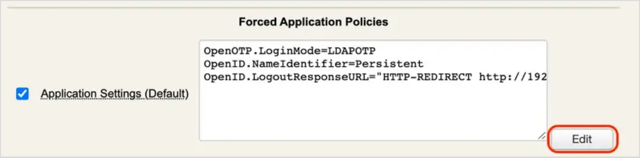
☑ Client Name Aliases zabbix

Comma-separated list of alternative client IDs.

☐ UID Attributes

Restricted list of LDAP login attributes replacing the attributes configured via uid_attrs in webadm.conf.

Click **EDIT** on **Application Settings (Default) :**

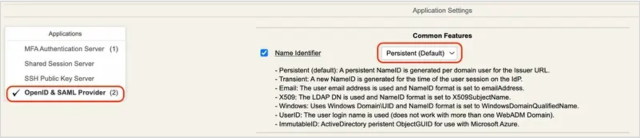


Forced Application Policies

☑ Application Settings (Default)

```
OpenOTP.LoginMode=LDAPOTP
OpenID.NameIdentifier=Persistent
OpenID.LogoutResponseURL="HTTP-REDIRECT http://192
```

Set **Name Identifier** to **Persistent :**



Application Settings

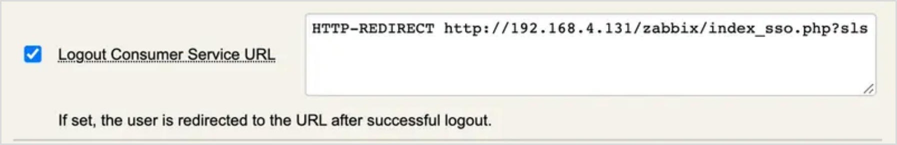
Common Features

☑ Name Identifier Persistent (Default)

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

Configure **Logout Consumer Service URL** to redirect user after successful logout :

```
HTTP-REDIRECT http://server_ip_or_name/zabbix/index_sso.php?sls
```

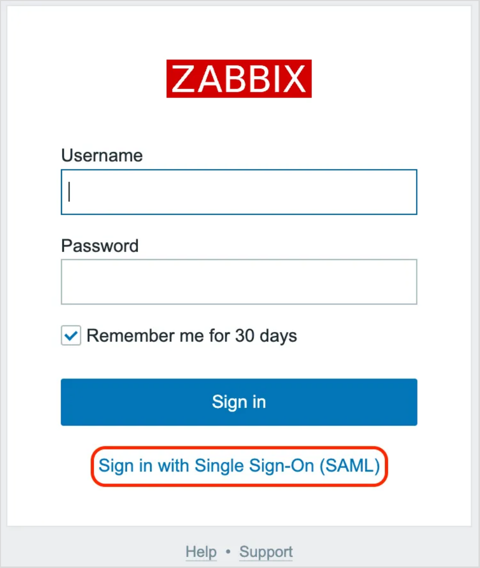


☑ Logout Consumer Service URL

```
HTTP-REDIRECT http://192.168.4.131/zabbix/index_sso.php?sls
```

If set, the user is redirected to the URL after successful logout.

> **SSO Authentication:**
>
> Go to: **http://server_ip_or_name/zabbix/**
>
> Click on **Sign in with Single Sign-On (saml)**

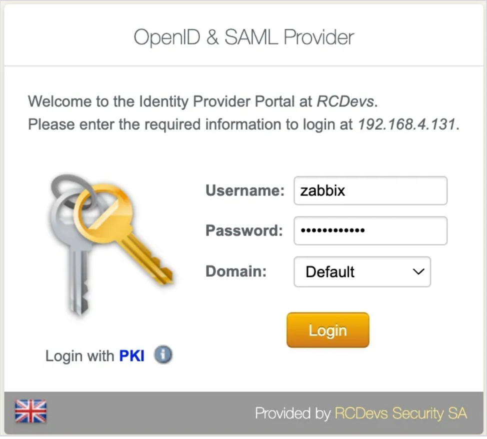


Username

Password

☑ Remember me for 30 days

Login with your user created in WebADM/Zabbix :



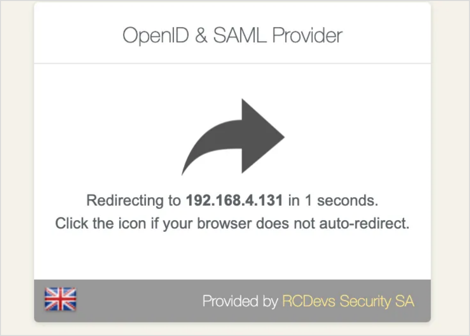
Welcome to the Identity Provider Portal at *RCDevs*.
Please enter the required information to login at *192.168.4.131*.

Username: zabbix

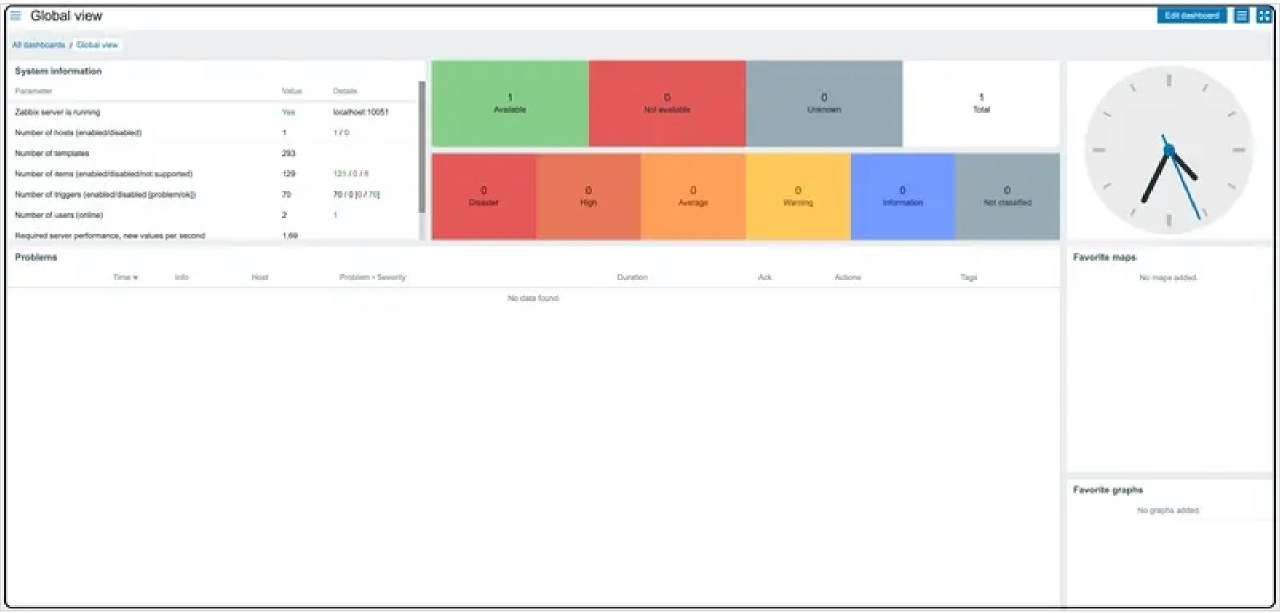
Password: ••••••••••••

Domain: Default

After **Successful Authentication** you are redirected to the Zabbix SP :



Redirecting to **192.168.4.131** in 1 seconds.
Click the icon if your browser does not auto-redirect.



System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	293	
Number of items (enabled/disabled/not supported)	129	121 / 0 / 8
Number of triggers (enabled/disabled [problem/ok])	70	70 / 0 [0 / 70]
Number of users (online)	2	1
Required server performance, new values per second	1.69	

1 Available	0 Not available	0 Unknown	1 Total

0 Disaster	0 High	0 Average	0 Warning	0 Information	0 Not classified

Problems

Time ▾	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags

No data found.

Favorite maps

No maps added.

Favorite graphs

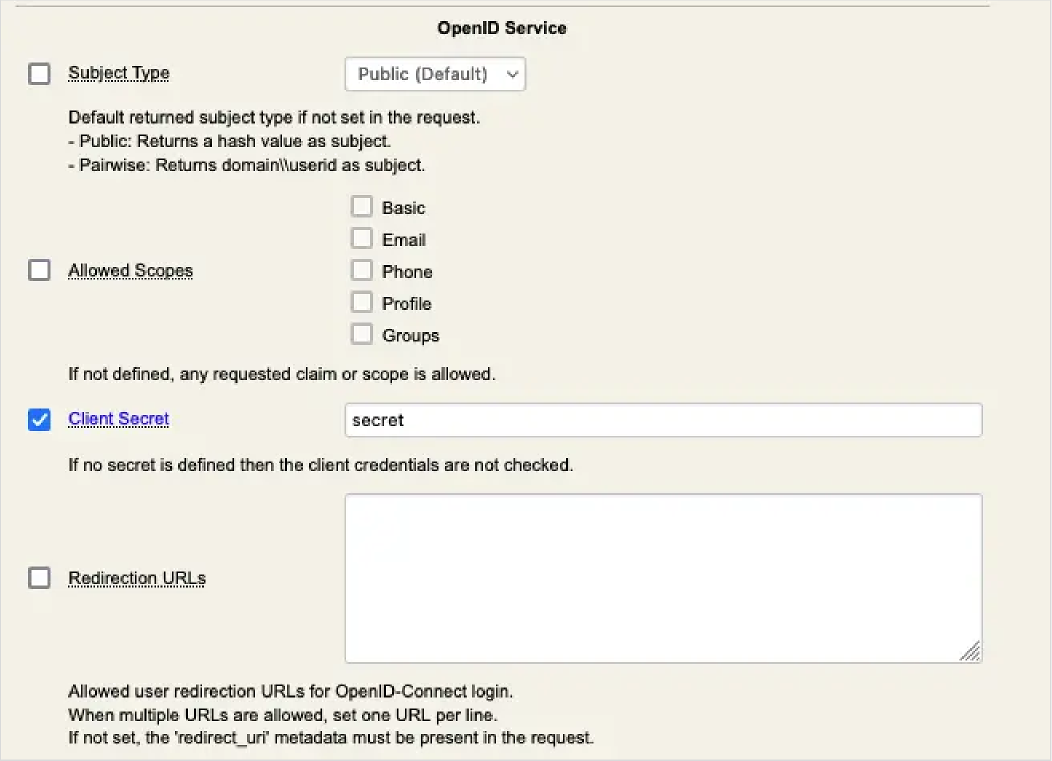
No graphs added.

3.3.10 WordPress (OIDC and SAML)

This was tested with WordPress 6.0.

3.3.10.1 Using OIDC

In WebADM, create a client policy named WordPress, and configure a secret for OpenID in OpenID Service settings:



OpenID Service

☐ Subject Type — Public (Default)

Default returned subject type if not set in the request.
- Public: Returns a hash value as subject.
- Pairwise: Returns domain\\userid as subject.

☐ Allowed Scopes — ☐ Basic ☐ Email ☐ Phone ☐ Profile ☐ Groups

If not defined, any requested claim or scope is allowed.

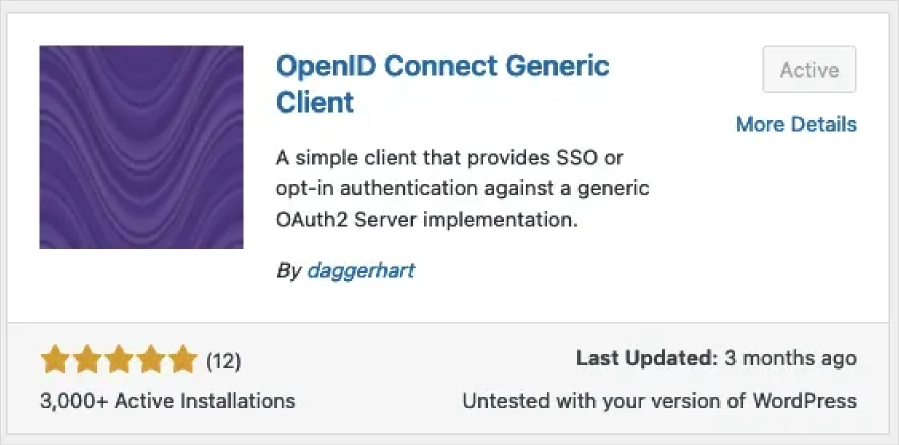
☑ Client Secret — secret

If no secret is defined then the client credentials are not checked.

☐ Redirection URLs

Allowed user redirection URLs for OpenID-Connect login.
When multiple URLs are allowed, set one URL per line.
If not set, the 'redirect_uri' metadata must be present in the request.

On WordPress, install and activate OpenID Connect Generic Client plugin:



OpenID Connect Generic Client

Active

More Details

A simple client that provides SSO or opt-in authentication against a generic OAuth2 Server implementation.

By daggerhart

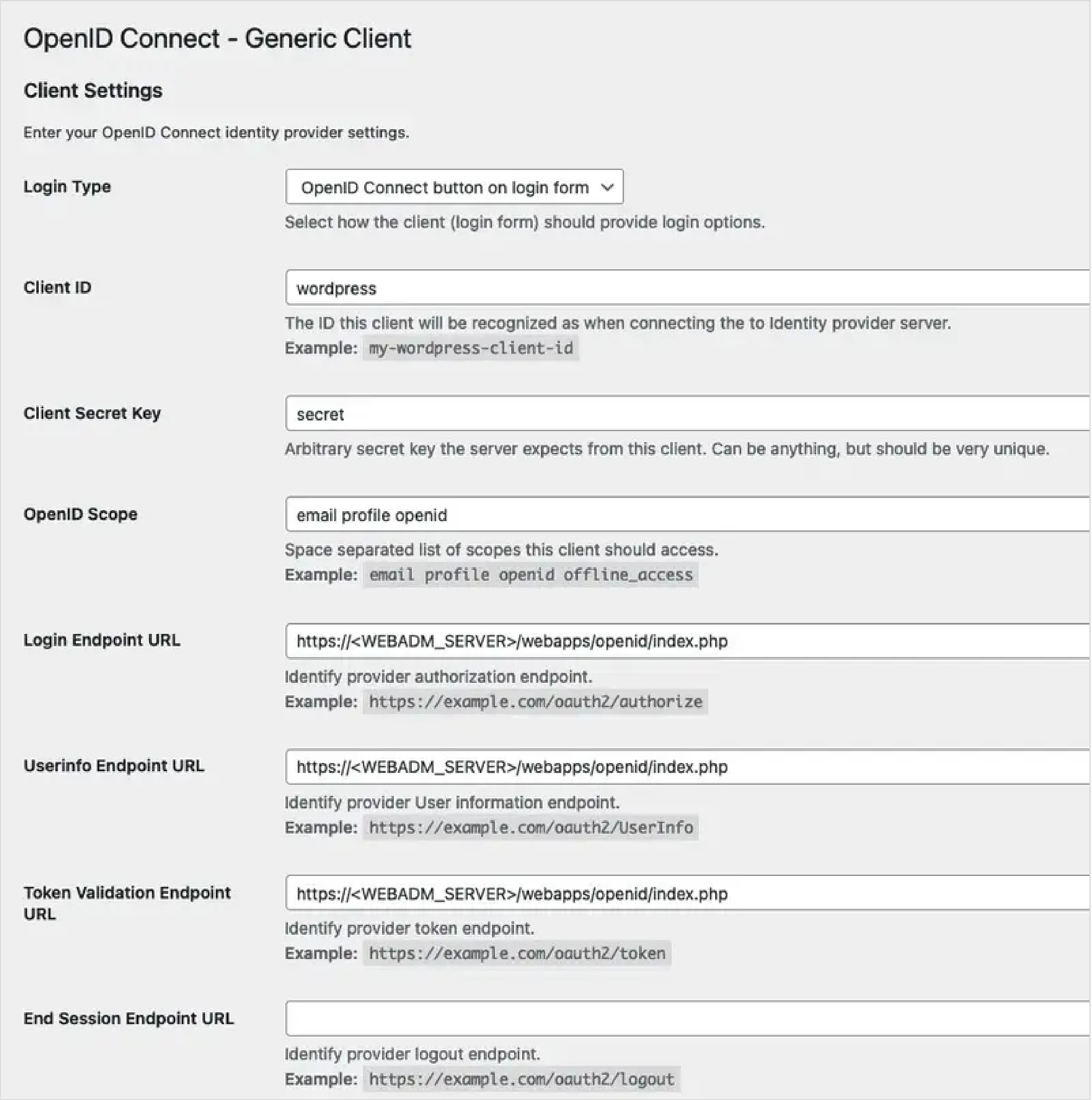
★★★★★ (12)

3,000+ Active Installations

Last Updated: 3 months ago

Untested with your version of WordPress

On WordPress, go to Settings->OpenID Connect Client menu, then configure the plugin (replace <WEBADM_SERVER> with actual IP or DNS of your setup):



OpenID Connect - Generic Client

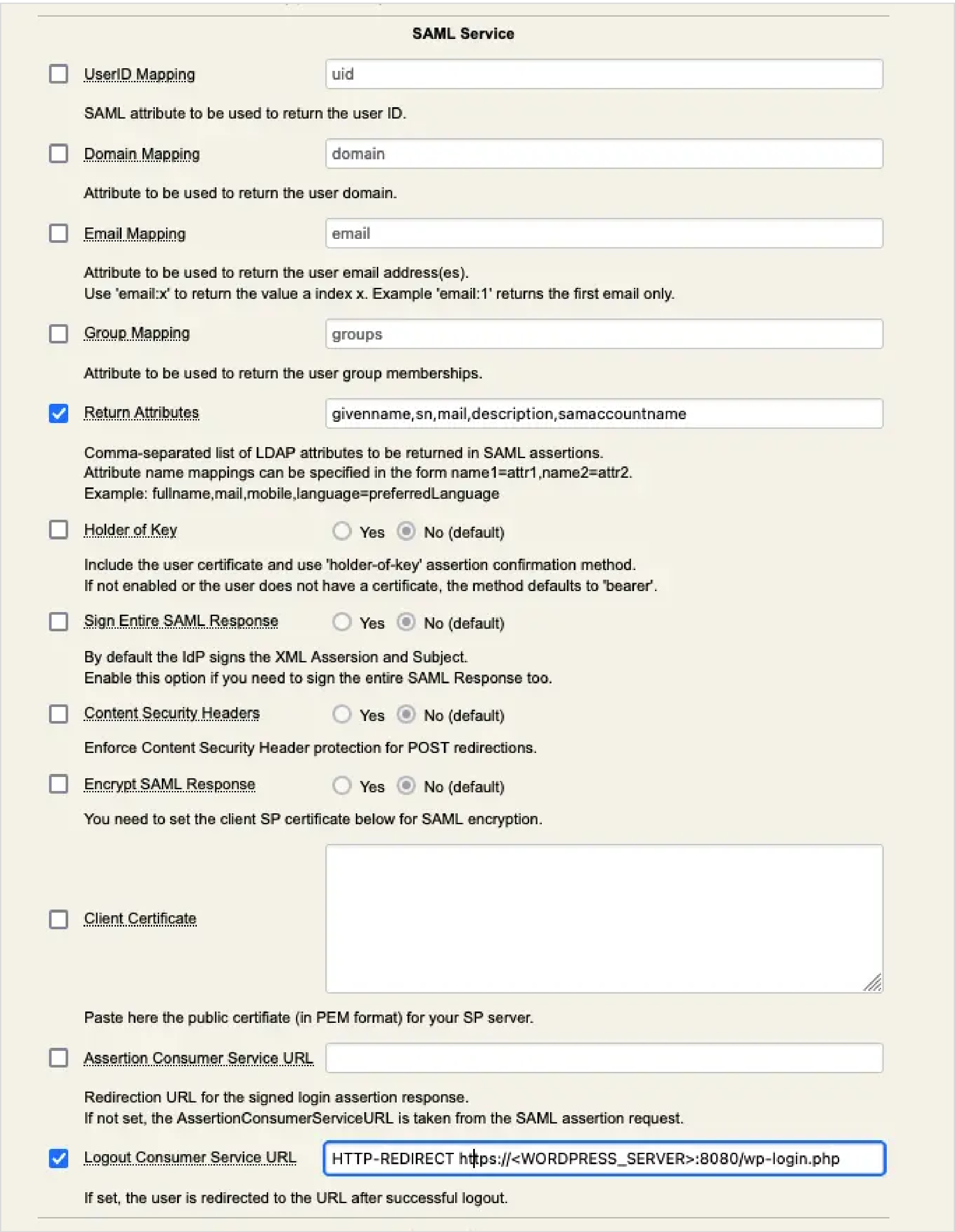
Client Settings

Enter your OpenID Connect identity provider settings.

Login Type	OpenID Connect button on login form Select how the client (login form) should provide login options.
Client ID	wordpress The ID this client will be recognized as when connecting the to Identity provider server. **Example:** `my-wordpress-client-id`
Client Secret Key	secret Arbitrary secret key the server expects from this client. Can be anything, but should be very unique.
OpenID Scope	email profile openid Space separated list of scopes this client should access. **Example:** `email profile openid offline_access`
Login Endpoint URL	https://<WEBADM_SERVER>/webapps/openid/index.php Identify provider authorization endpoint. **Example:** `https://example.com/oauth2/authorize`
Userinfo Endpoint URL	https://<WEBADM_SERVER>/webapps/openid/index.php Identify provider User information endpoint. **Example:** `https://example.com/oauth2/UserInfo`
Token Validation Endpoint URL	https://<WEBADM_SERVER>/webapps/openid/index.php Identify provider token endpoint. **Example:** `https://example.com/oauth2/token`
End Session Endpoint URL	 Identify provider logout endpoint. **Example:** `https://example.com/oauth2/logout`

3.3.10.2 Using SAML

In WebADM, create a client policy named WordPress, and configure following SAML settings (<WORDPRESS_SERVER:8080> must be changed to fit your setup):



SAML Service

☐ UserID Mapping: uid
SAML attribute to be used to return the user ID.

☐ Domain Mapping: domain
Attribute to be used to return the user domain.

☐ Email Mapping: email
Attribute to be used to return the user email address(es).
Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

☐ Group Mapping: groups
Attribute to be used to return the user group memberships.

☑ Return Attributes: givenname,sn,mail,description,samaccountname
Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

☐ Holder of Key: ○ Yes ● No (default)
Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

☐ Sign Entire SAML Response: ○ Yes ● No (default)
By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

☐ Content Security Headers: ○ Yes ● No (default)
Enforce Content Security Header protection for POST redirections.

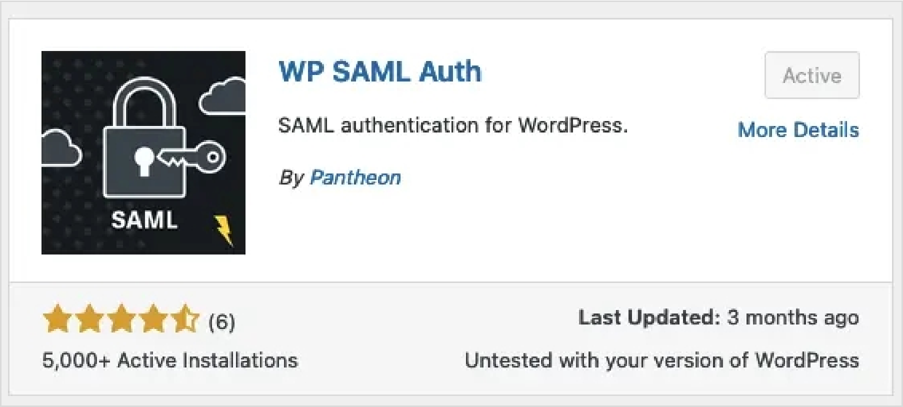
☐ Encrypt SAML Response: ○ Yes ● No (default)
You need to set the client SP certificate below for SAML encryption.

☐ Client Certificate:
Paste here the public certifiate (in PEM format) for your SP server.

☐ Assertion Consumer Service URL:
Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

☑ Logout Consumer Service URL: HTTP-REDIRECT https://<WORDPRESS_SERVER>:8080/wp-login.php
If set, the user is redirected to the URL after successful logout.

On WordPress, install and activate OpenID Connect Generic Client plugin:



WP SAML Auth

SAML authentication for WordPress.

By Pantheon

Active

More Details

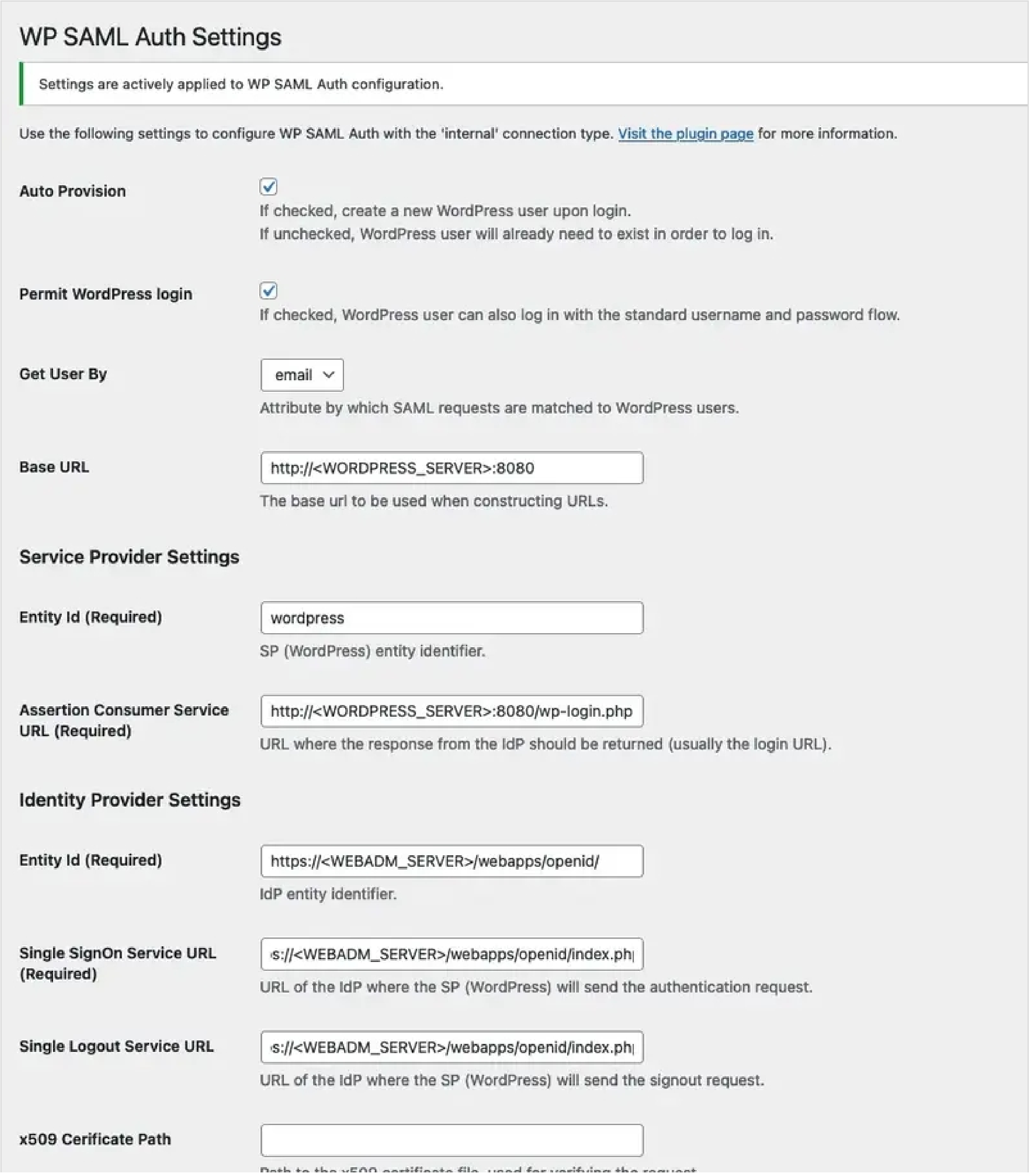
★★★★⯨ (6)

5,000+ Active Installations

Last Updated: 3 months ago

Untested with your version of WordPress

On WordPress, go to Settings->WP SAML Auth menu, then configure the plugin (replace <WEBADM_SERVER> with actual IP or DNS of your setup):



WP SAML Auth Settings

Settings are actively applied to WP SAML Auth configuration.

Use the following settings to configure WP SAML Auth with the 'internal' connection type. Visit the plugin page for more information.

Auto Provision ☑
If checked, create a new WordPress user upon login.
If unchecked, WordPress user will already need to exist in order to log in.

Permit WordPress login ☑
If checked, WordPress user can also log in with the standard username and password flow.

Get User By email
Attribute by which SAML requests are matched to WordPress users.

Base URL http://<WORDPRESS_SERVER>:8080
The base url to be used when constructing URLs.

Service Provider Settings

Entity Id (Required) wordpress
SP (WordPress) entity identifier.

Assertion Consumer Service URL (Required) http://<WORDPRESS_SERVER>:8080/wp-login.php
URL where the response from the IdP should be returned (usually the login URL).

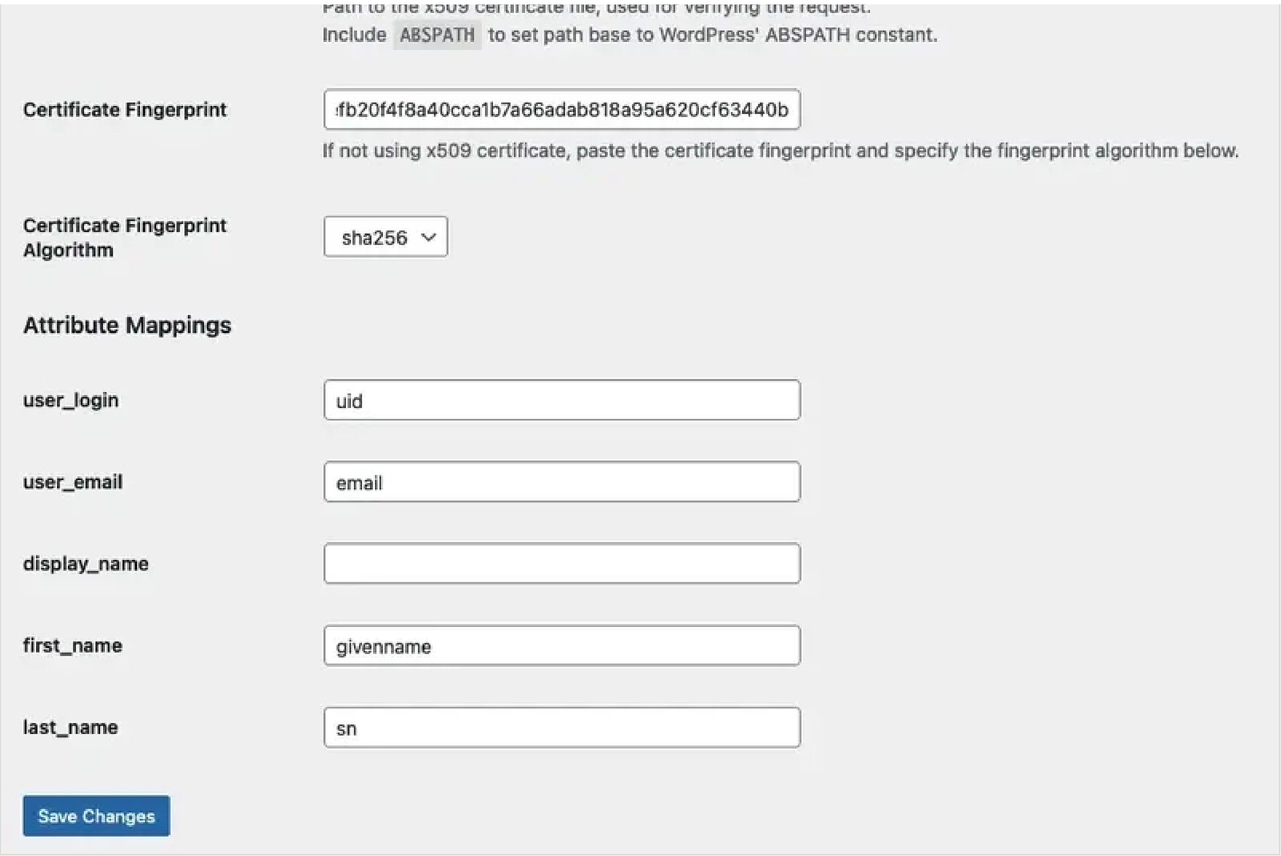
Identity Provider Settings

Entity Id (Required) https://<WEBADM_SERVER>/webapps/openid/
IdP entity identifier.

Single SignOn Service URL (Required) s://<WEBADM_SERVER>/webapps/openid/index.ph
URL of the IdP where the SP (WordPress) will send the authentication request.

Single Logout Service URL s://<WEBADM_SERVER>/webapps/openid/index.ph
URL of the IdP where the SP (WordPress) will send the signout request.

x509 Cerificate Path
Path to the x509 certificate file, used for verifying the request.



Path to the x509 certificate file, used for verifying the request.
Include ABSPATH to set path base to WordPress' ABSPATH constant.

Certificate Fingerprint: fb20f4f8a40cca1b7a66adab818a95a620cf63440b

If not using x509 certificate, paste the certificate fingerprint and specify the fingerprint algorithm below.

Certificate Fingerprint Algorithm: sha256

Attribute Mappings

user_login: uid

user_email: email

display_name:

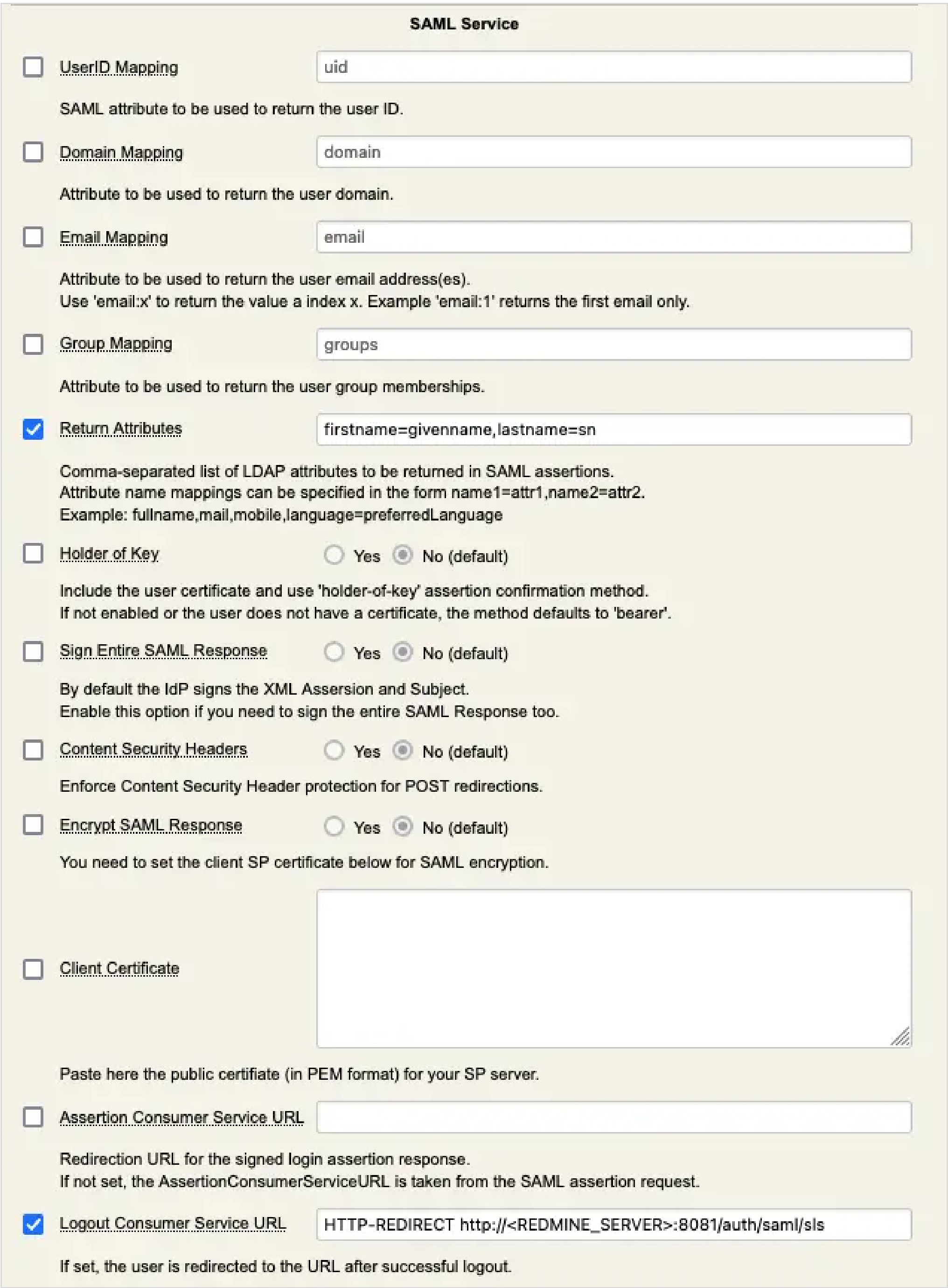
first_name: givenname

last_name: sn

3.3.11 Redmine (SAML)

This was tested with Redmine 5.0.1.

In WebADM, create a client policy named redmine, and configure following SAML settings (<REDMINE_SERVER:8081> must be changed to fit your setup):



SAML Service

☐ UserID Mapping: uid

SAML attribute to be used to return the user ID.

☐ Domain Mapping: domain

Attribute to be used to return the user domain.

☐ Email Mapping: email

Attribute to be used to return the user email address(es).
Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

☐ Group Mapping: groups

Attribute to be used to return the user group memberships.

☑ Return Attributes: firstname=givenname,lastname=sn

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

☐ Holder of Key ○ Yes ● No (default)

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

☐ Sign Entire SAML Response ○ Yes ● No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

☐ Content Security Headers ○ Yes ● No (default)

Enforce Content Security Header protection for POST redirections.

☐ Encrypt SAML Response ○ Yes ● No (default)

You need to set the client SP certificate below for SAML encryption.

☐ Client Certificate

Paste here the public certifiate (in PEM format) for your SP server.

☐ Assertion Consumer Service URL

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

☑ Logout Consumer Service URL: HTTP-REDIRECT http://<REDMINE_SERVER>:8081/auth/saml/sls

If set, the user is redirected to the URL after successful logout.

In redmine server, follow these steps to install Redmine OmniAuth SAML plugin from AlphaNodes/redmine_saml repository (assumes that you are at the root of your redmine folder):

```
git clone https://github.com/alphanodes/additionals.git plugins/additionals
git clone https://github.com/alphanodes/redmine_saml.git plugins/redmine_saml
cp plugins/redmine_saml/sample-saml-initializers.rb config/initializers/saml.rb
```

Then, edit config/initializers/saml.rb and adapt settings to your setup (replace <WEBADM_SERVER> and <REDMINE_SERVER> values):

```
require Rails.root.join('plugins/redmine_saml/lib/redmine_saml')
require Rails.root.join('plugins/redmine_saml/lib/redmine_saml/base')
RedmineSaml::Base.configure do |config|
  config.saml = {
    sp_entity_id: 'redmine',
    idp_sso_service_url: 'https://<WEBADM_SERVER>/webapps/openid/index.php',
    assertion_consumer_service_url: 'https://<REDMINE_SERVER>/auth/saml/callback',
    issuer: 'https://<REDMINE_SERVER>/auth/saml/metadata',
    single_logout_service_url: 'https://<REDMINE_SERVER>/auth/saml/sls',
    idp_sso_target_url: 'https://<WEBADM_SERVER>/webapps/openid/openotp.php',
    idp_cert_fingerprint: '0fb6a5f22dd609d9364d45846bdd4afd2e3f52f3',
    name_identifier_format: 'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent',
    signout_url: 'https://<WEBADM_SERVER>/webapps/openid/index.php',
    idp_slo_target_url: 'https://<WEBADM_SERVER>/webapps/openid/index.php',
    name_identifier_value: 'mail',
    attribute_mapping: {
      login: 'extra|raw_info|username',
      mail: 'extra|raw_info|email',
      firstname: 'extra|raw_info|firstname',
      lastname: 'extra|raw_info|lastname',
      admin: 'extra|raw_info|admin'
    }
  }
  config.on_login do |omniauth_hash, user|
  end
end
```

Finally, install dependencies and install plugin:

```
bundle install
bundle exec rake redmine:plugins:migrate RAILS_ENV=production
```

Restart your Redmine server, then connected as admin in Redmine, go to Administration->Plugins->Configure of Redmine SAML menu, and enable `Create users automatically?` setting.

3.3.12 Splunk (SAML)

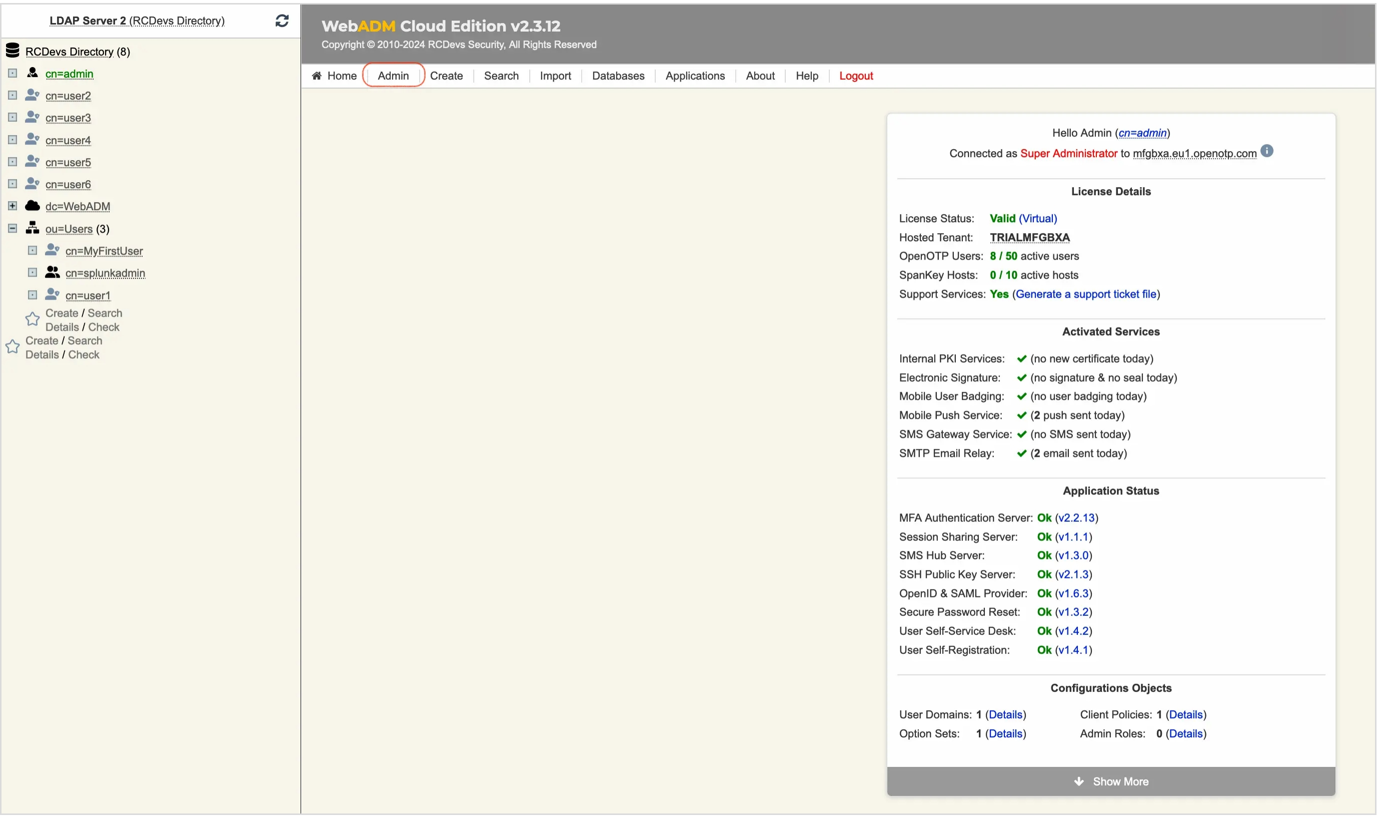
Splunk supports Security Assertion Markup Language (SAML) for single sign-on (SSO) integration.

Here are the general steps to integrate Splunk with SAML :

In WebADM, we need to:

- Configure a Client Policies (Splunk).
- Download the metadata for use on the Service Provider (SP).

- We also need the WebADM CA (Certificate Authority).



LDAP Server 2 (RCDevs Directory)

RCDevs Directory (8)
- cn=admin
- cn=user2
- cn=user3
- cn=user4
- cn=user5
- cn=user6
- dc=WebADM
- ou=Users (3)
  - cn=MyFirstUser
  - cn=splunkadmin
  - cn=user1
  - Create / Search Details / Check
- Create / Search Details / Check

WebADM Cloud Edition v2.3.12
Copyright © 2010-2024 RCDevs Security, All Rights Reserved

Home | Admin | Create | Search | Import | Databases | Applications | About | Help | Logout

Hello Admin (cn=admin)
Connected as Super Administrator to mfgbxa.eu1.openotp.com

License Details

License Status: Valid (Virtual)
Hosted Tenant: TRIALMFGBXA
OpenOTP Users: 8 / 50 active users
SpanKey Hosts: 0 / 10 active hosts
Support Services: Yes (Generate a support ticket file)

Activated Services

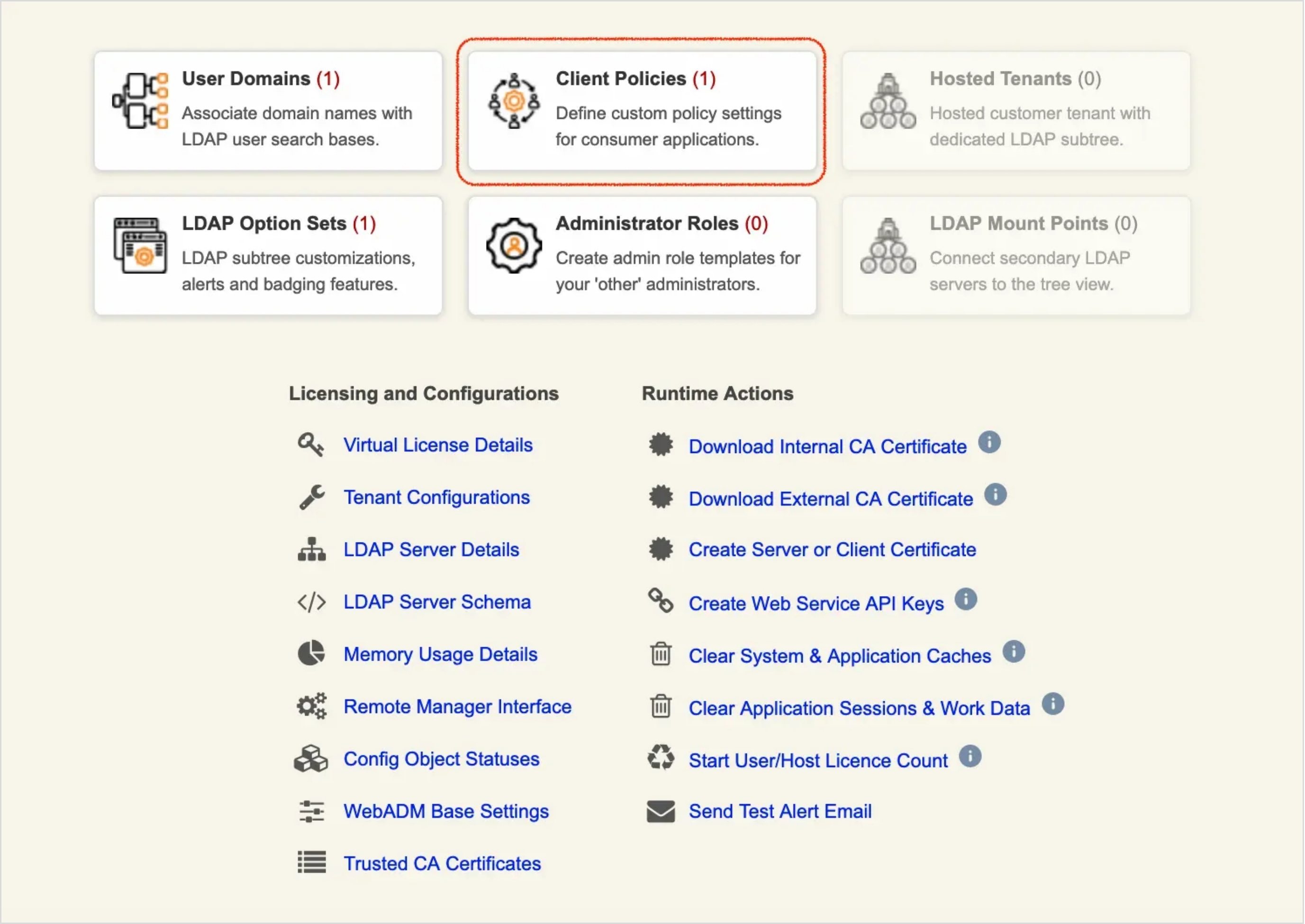
Internal PKI Services: ✓ (no new certificate today)
Electronic Signature: ✓ (no signature & no seal today)
Mobile User Badging: ✓ (no user badging today)
Mobile Push Service: ✓ (2 push sent today)
SMS Gateway Service: ✓ (no SMS sent today)
SMTP Email Relay: ✓ (2 email sent today)

Application Status

MFA Authentication Server: Ok (v2.2.13)
Session Sharing Server: Ok (v1.1.1)
SMS Hub Server: Ok (v1.3.0)
SSH Public Key Server: Ok (v2.1.3)
OpenID & SAML Provider: Ok (v1.6.3)
Secure Password Reset: Ok (v1.3.2)
User Self-Service Desk: Ok (v1.4.2)
User Self-Registration: Ok (v1.4.1)

Configurations Objects

User Domains: 1 (Details) Client Policies: 1 (Details)
Option Sets: 1 (Details) Admin Roles: 0 (Details)



User Domains (1)
Associate domain names with LDAP user search bases.

Client Policies (1)
Define custom policy settings for consumer applications.

Hosted Tenants (0)
Hosted customer tenant with dedicated LDAP subtree.

LDAP Option Sets (1)
LDAP subtree customizations, alerts and badging features.

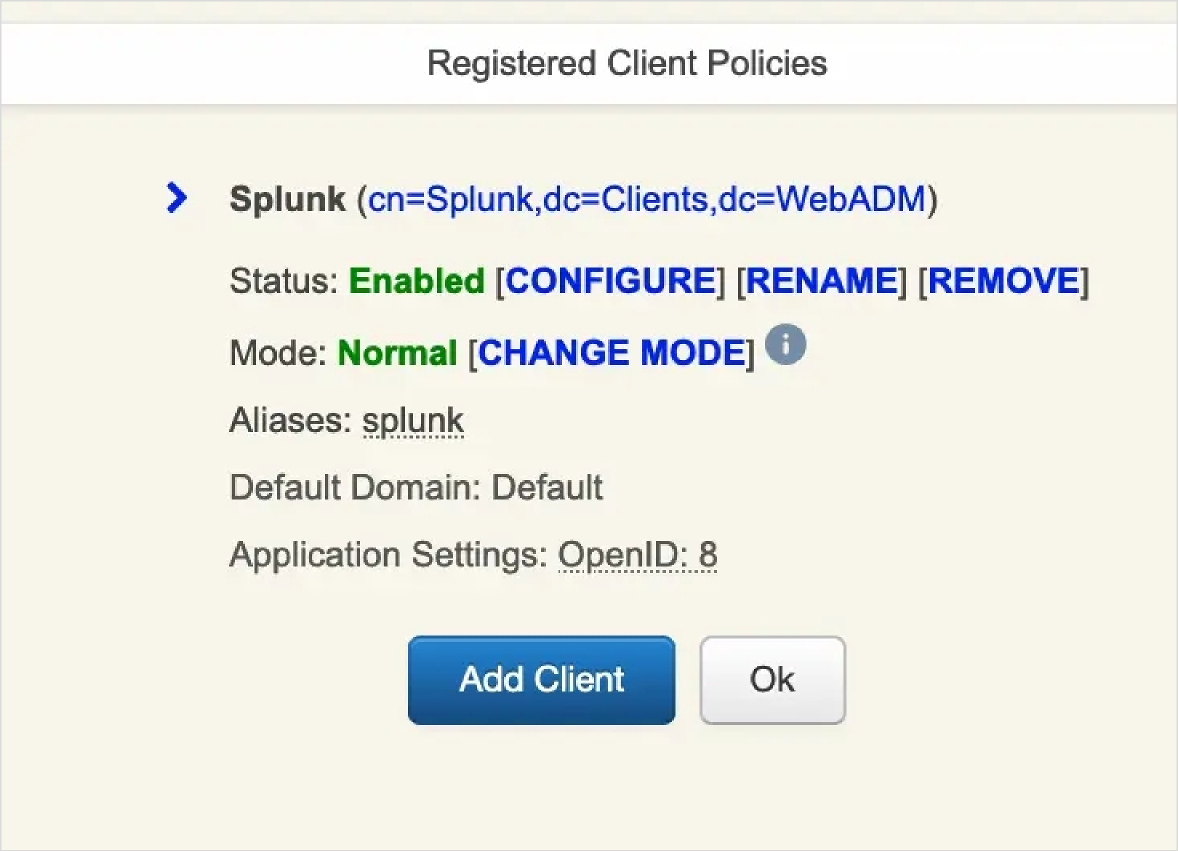
Administrator Roles (0)
Create admin role templates for your 'other' administrators.

LDAP Mount Points (0)
Connect secondary LDAP servers to the tree view.

Licensing and Configurations
- Virtual License Details
- Tenant Configurations
- LDAP Server Details
- LDAP Server Schema
- Memory Usage Details
- Remote Manager Interface
- Config Object Statuses
- WebADM Base Settings
- Trusted CA Certificates

Runtime Actions
- Download Internal CA Certificate
- Download External CA Certificate
- Create Server or Client Certificate
- Create Web Service API Keys
- Clear System & Application Caches
- Clear Application Sessions & Work Data
- Start User/Host Licence Count
- Send Test Alert Email

We will name the Client Policies : **Splunk**



Registered Client Policies

> Splunk (cn=Splunk,dc=Clients,dc=WebADM)

Status: Enabled [CONFIGURE] [RENAME] [REMOVE]

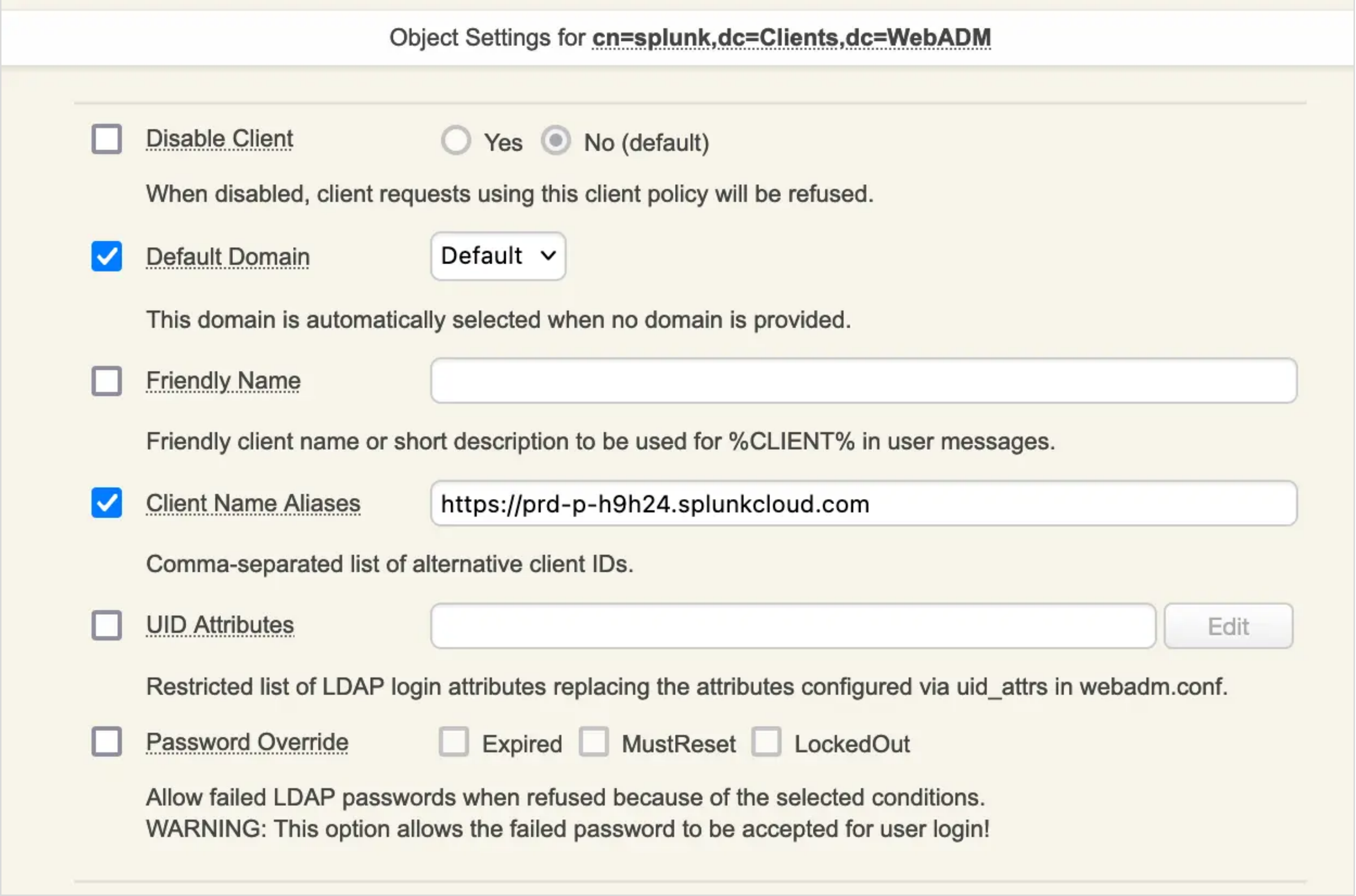
Mode: Normal [CHANGE MODE]

Aliases: splunk

Default Domain: Default

Application Settings: OpenID: 8

Client Name Aliases, It’s the link with which you connect to SplunkCloud. We will use it later in the SAML configuration for Entity ID(SP).



Object Settings for cn=splunk,dc=Clients,dc=WebADM

□ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☑ Default Domain Default

This domain is automatically selected when no domain is provided.

□ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases https://prd-p-h9h24.splunkcloud.com

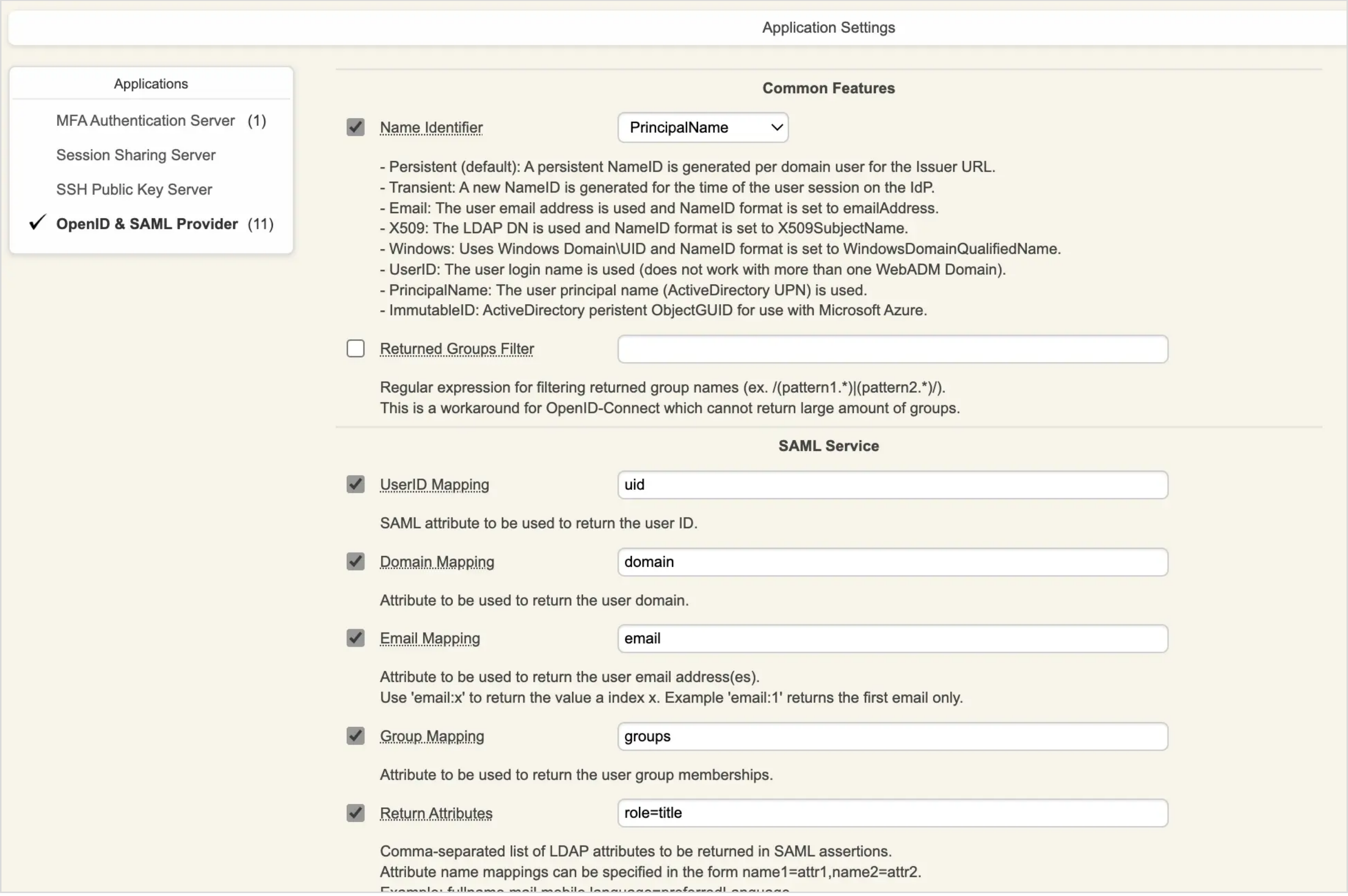
Comma-separated list of alternative client IDs.

□ UID Attributes

Restricted list of LDAP login attributes replacing the attributes configured via uid_attrs in webadm.conf.

□ Password Override □ Expired □ MustReset □ LockedOut

Allow failed LDAP passwords when refused because of the selected conditions.
WARNING: This option allows the failed password to be accepted for user login!



Common Features

☑ Name Identifier: PrincipalName

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- PrincipalName: The user principal name (ActiveDirectory UPN) is used.
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

☐ Returned Groups Filter

Regular expression for filtering returned group names (ex. /(pattern1.*)|(pattern2.*)/).
This is a workaround for OpenID-Connect which cannot return large amount of groups.

SAML Service

☑ UserID Mapping: uid

SAML attribute to be used to return the user ID.

☑ Domain Mapping: domain

Attribute to be used to return the user domain.

☑ Email Mapping: email

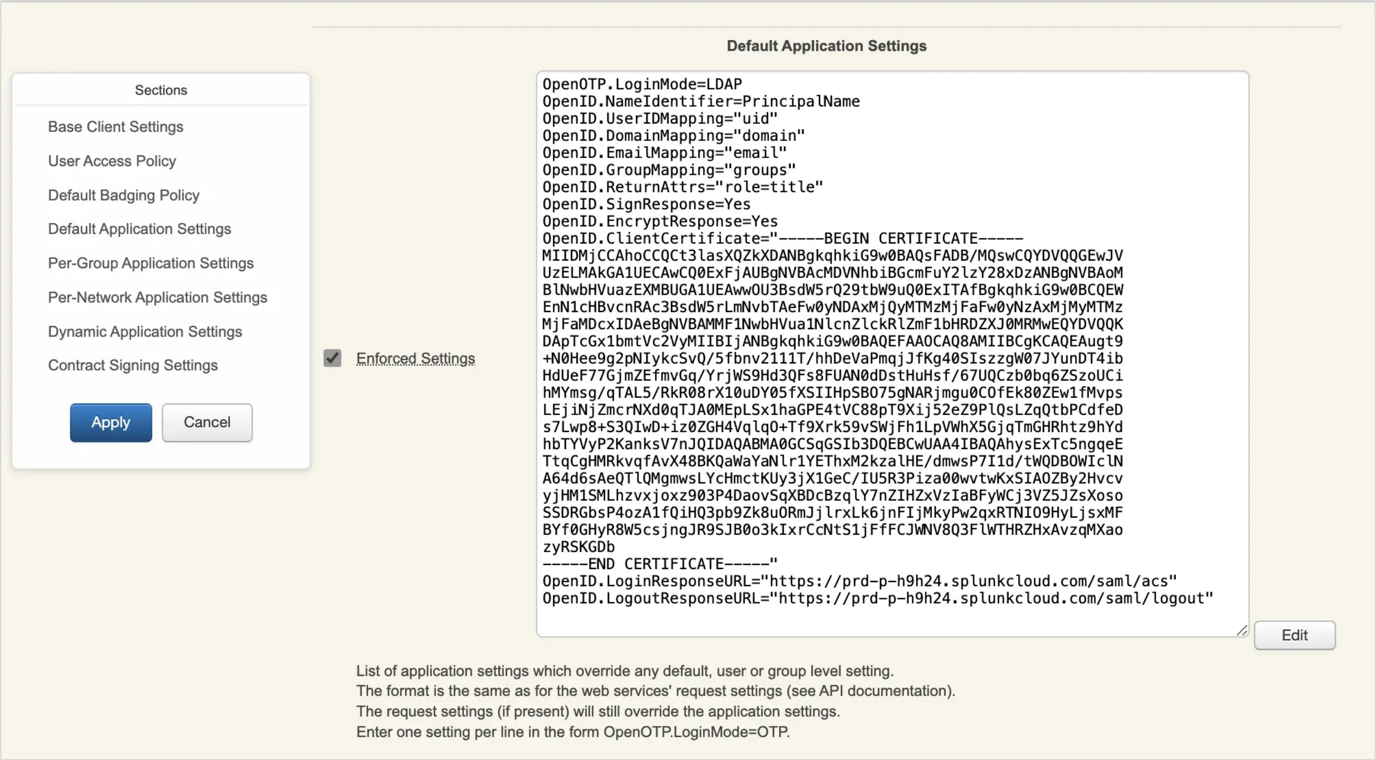
Attribute to be used to return the user email address(es).
Use 'email:x' to return the value a index x. Example 'email:1' returns the first email only.

☑ Group Mapping: groups

Attribute to be used to return the user group memberships.

☑ Return Attributes: role=title

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage


Sections
Base Client Settings
User Access Policy
Default Badging Policy
Default Application Settings
Per-Group Application Settings
Per-Network Application Settings
Dynamic Application Settings
Contract Signing Settings
Apply
Cancel
Edit

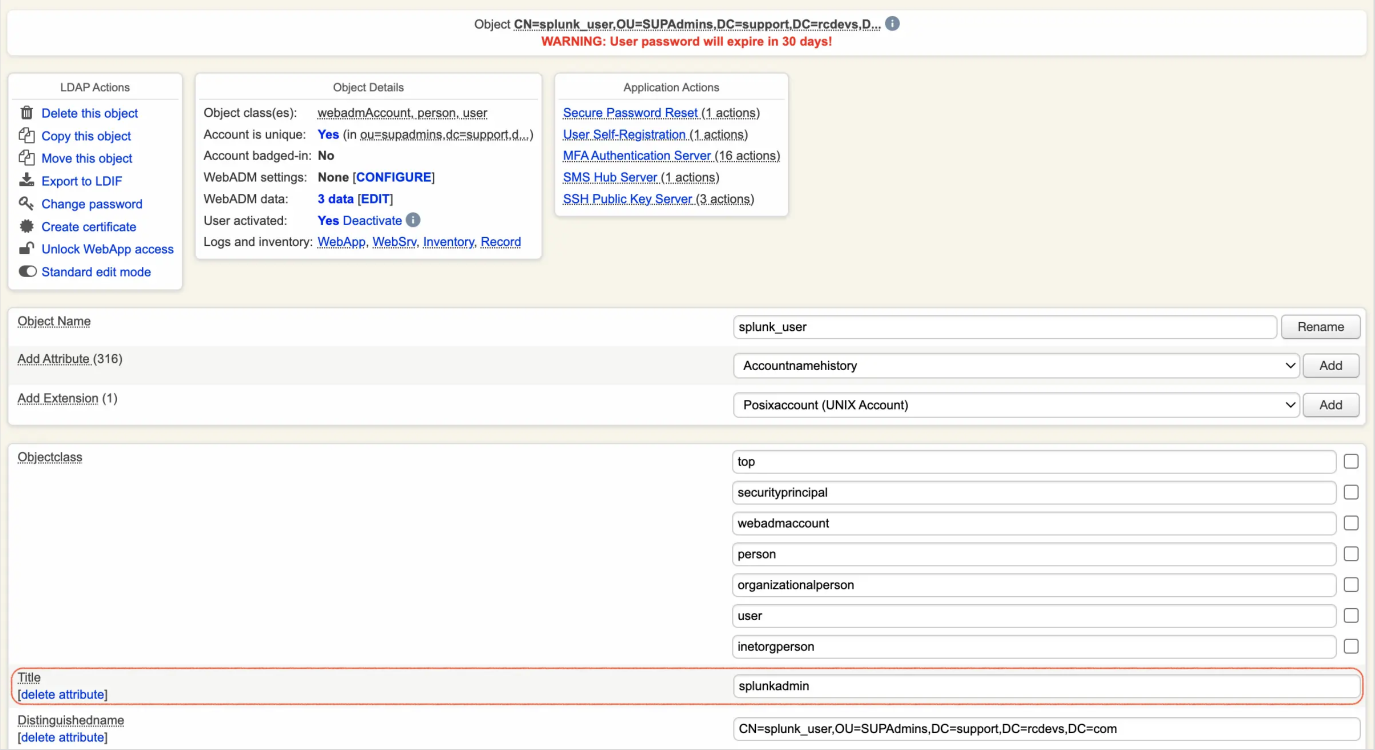
Default Application Settings

☑ Enforced Settings

```
OpenOTP.LoginMode=LDAP
OpenID.NameIdentifier=PrincipalName
OpenID.UserIDMapping="uid"
OpenID.DomainMapping="domain"
OpenID.EmailMapping="email"
OpenID.GroupMapping="groups"
OpenID.ReturnAttrs="role=title"
OpenID.SignResponse=Yes
OpenID.EncryptResponse=Yes
OpenID.ClientCertificate="-----BEGIN CERTIFICATE-----
MIIDMjCCAhoCCQCt3lasXQZkXDANBgkqhkiG9w0BAQsFADB/MQswCQYDVQQGEwJV
UzELMAkGA1UECAwCQ0ExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDzANBgNVBAoM
BlNwbHVuazEXMBUGA1UEAwwOU3BsdW5rQ29tbW9uQ0ExITAfBgkqhkiG9w0BCQEW
EnN1cHBvcnRAc3BsdW5rLmNvbTAeFw0yNDAxMjQyMTMzMjFaFw0yNzAxMjMyMTMz
MjFaMDcxIDAeBgNVBAMMF1NwbHVua1NlcnZlckRlZmF1bHRDZXJ0MRMwEQYDVQQK
DApTcGx1bmtVc2VyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAugt9
+N0Hee9g2pNIykcSvQ/5fbnv2111T/hhDeVaPmqjJfKg40SIszzgW07JYunDT4ib
HdUeF77GjmZEfmvGq/YrjWS9Hd3QFs8FUAN0dDstHuHsf/67UQCzb0bq6ZSzoUCi
hMYmsg/qTAL5/RkR08rX10uDY05fXSIIHpSBO75gNARjmgu0COfEk80ZEw1fMvps
LEjiNjZmcrNXd0qTJA0MEpLSx1haGPE4tVC88pT9Xij52eZ9PlQsLZqQtbPCdfeD
s7Lwp8+S3QIwD+iz0ZGH4VqlqO+Tf9Xrk59vSWjFh1LpVWhX5GjqTmGHRhtz9hYd
hbTYVyP2KanksV7nJQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAhysExTc5ngqeE
TtqCgHMRkvqfAvX48BKQaWaYaNlr1YEThxM2kzalHE/dmwsP7I1d/tWQDBOWIclN
A64d6sAeQTlQMgmwsLYcHmctKUy3jX1GeC/IU5R3Piza00wvtwKxSIAOZBy2Hvcv
yjHM1SMLhzvxjoxz903P4DaovSqXBDcBzqlY7nZIHZxVzIaBFyWCj3VZ5JZsXoso
SSDRGbsP4ozA1fQiHQ3pb9Zk8uORmJjlrxLk6jnFIjMkyPw2qxRTNIO9HyLjsxMF
BYf0GHyR8W5csjngJR9SJB0o3kIxrCcNtS1jFfFCJWNV8Q3FlWTHRZHxAvzqMXao
zyRSKGDb
-----END CERTIFICATE-----"
OpenID.LoginResponseURL="https://prd-p-h9h24.splunkcloud.com/saml/acs"
OpenID.LogoutResponseURL="https://prd-p-h9h24.splunkcloud.com/saml/logout"
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

In Splunk, a user must have a role within a group. Therefore, we need to add the Title attribute and assign it the value splunkadmin (which is a group already created in Splunk). You may have noticed that we configured the “Return Attributes” before: role=title. This means that for our user “splunk_user” the “splunkadmin” role will be sent to SP in the SAML response.



Object CN=splunk_user,OU=SUPAdmins,DC=support,DC=rcdevs,D...
WARNING: User password will expire in 30 days!

LDAP Actions
- Delete this object
- Copy this object
- Move this object
- Export to LDIF
- Change password
- Create certificate
- Unlock WebApp access
- Standard edit mode

Object Details

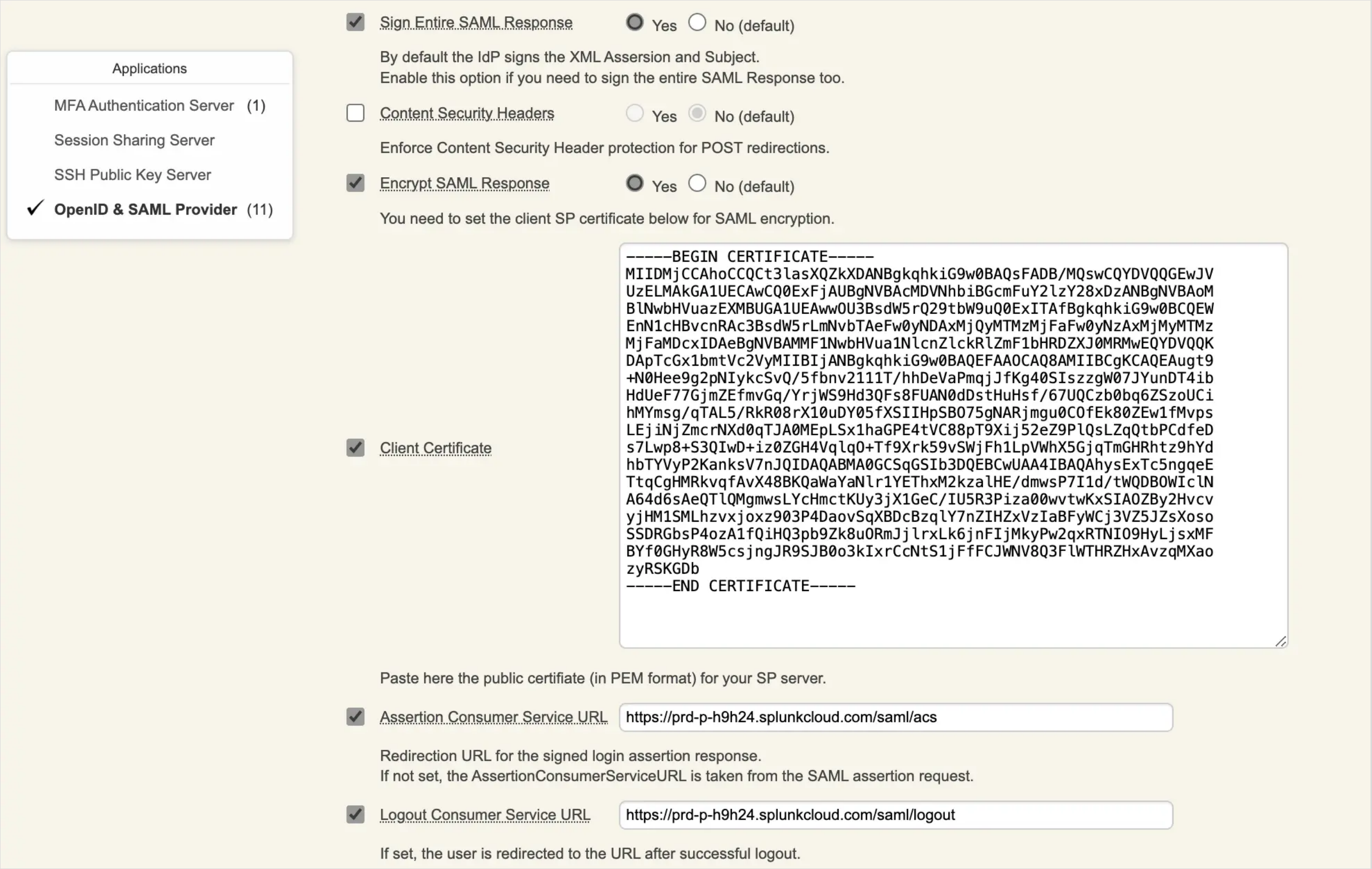
Object class(es):	webadmAccount, person, user
Account is unique:	Yes (in ou=supadmins,dc=support,d...)
Account badged-in:	No
WebADM settings:	None [CONFIGURE]
WebADM data:	3 data [EDIT]
User activated:	Yes Deactivate
Logs and inventory:	WebApp, WebSrv, Inventory, Record

Application Actions
- Secure Password Reset (1 actions)
- User Self-Registration (1 actions)
- MFA Authentication Server (16 actions)
- SMS Hub Server (1 actions)
- SSH Public Key Server (3 actions)

Object Name	splunk_user
Add Attribute (316)	Accountnamehistory
Add Extension (1)	Posixaccount (UNIX Account)

Objectclass	top ☐ securityprincipal ☐ webadmaccount ☐ person ☐ organizationalperson ☐ user ☐ inetorgperson ☐
Title [delete attribute]	splunkadmin
Distinguishedname [delete attribute]	CN=splunk_user,OU=SUPAdmins,DC=support,DC=rcdevs,DC=com

Here, we will put the certificate and other configurations found in the metadata file of the SP. For the certificate, it needs to be in PEM format.



Applications
- MFA Authentication Server (1)
- Session Sharing Server
- SSH Public Key Server
- ✓ **OpenID & SAML Provider** (11)

☑ Sign Entire SAML Response ● Yes ○ No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

☐ Content Security Headers ○ Yes ● No (default)

Enforce Content Security Header protection for POST redirections.

☑ Encrypt SAML Response ● Yes ○ No (default)

You need to set the client SP certificate below for SAML encryption.

☑ Client Certificate

```
-----BEGIN CERTIFICATE-----
MIIDMjCCAhoCCQCt3lasXQZkXDANBgkqhkiG9w0BAQsFADB/MQswCQYDVQQGEwJV
UzELMAkGA1UECAwCQ0ExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDzANBgNVBAoM
BlNwbHVuazEXMBUGA1UEAwwOU3BsdW5rQ29tbW9uQ0ExITAfBgkqhkiG9w0BCQEW
EnN1cHBvcnRAc3BsdW5rLmNvbTAeFw0yNDAxMjQyMTMzMjFaFw0yNzAxMjMyMTMz
MjFaMDcxIDAeBgNVBAMMF1NwbHVua1NlcnZlckRlZmF1bHRDZXJ0MRMwEQYDVQQK
DApTcGx1bmtVc2VyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAugt9
+N0Hee9g2pNIykcSvQ/5fbnv2111T/hhDeVaPmqjJfKg40SIszzgW07JYunDT4ib
HdUeF77GjmZEfmvGq/YrjWS9Hd3QFs8FUAN0dDstHuHsf/67UQCzb0bq6ZSzoUCi
hMYmsg/qTAL5/RkR08rX10uDY05fXSIIHpSBO75gNARjmgu0COfEk80ZEw1fMvps
LEjiNjZmcrNXd0qTJA0MEpLSx1haGPE4tVC88pT9Xij52eZ9PlQsLZqQtbPCdfeD
s7Lwp8+S3QIwD+iz0ZGH4VqlqO+Tf9Xrk59vSWjFh1LpVWhX5GjqTmGHRhtz9hYd
hbTYVyP2KanksV7nJQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQAhysExTc5ngqeE
TtqCgHMRkvqfAvX48BKQaWaYaNlr1YEThxM2kzalHE/dmwsP7I1d/tWQDBOWIclN
A64d6sAeQTlQMgmwsLYcHmctKUy3jX1GeC/IU5R3Piza00wvtwKxSIAOZBy2Hvcv
yjHM1SMLhzvxjoxz903P4DaovSqXBDcBzqlY7nZIHZxVzIaBFyWCj3VZ5JZsXoso
SSDRGbsP4ozA1fQiHQ3pb9Zk8uORmJjlrxLk6jnFIjMkyPw2qxRTNIO9HyLjsxMF
BYf0GHyR8W5csjngJR9SJB0o3kIxrCcNtS1jFfFCJWNV8Q3FlWTHRZHxAvzqMXao
zyRSKGDb
-----END CERTIFICATE-----
```

Paste here the public certificate (in PEM format) for your SP server.

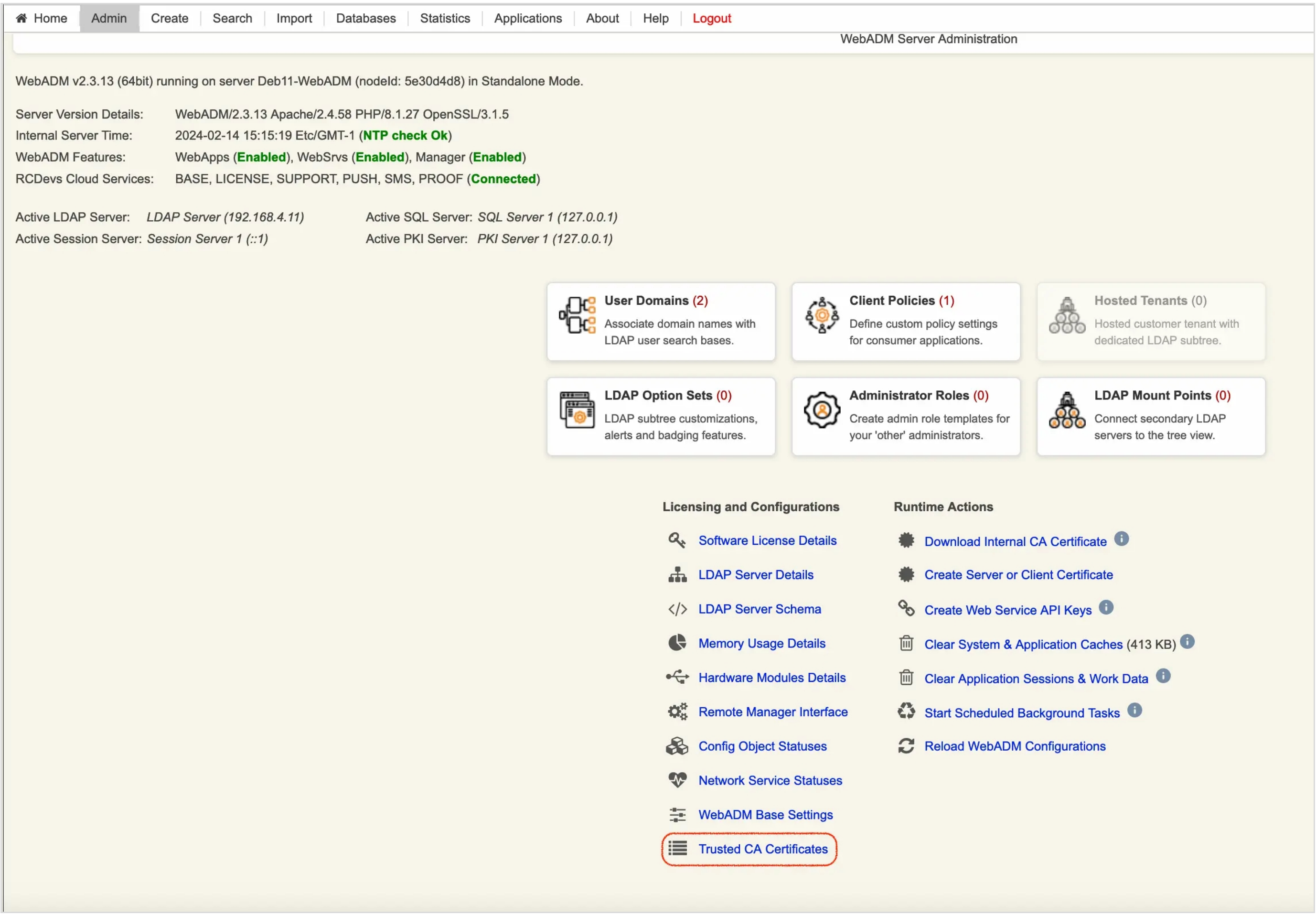
☑ Assertion Consumer Service URL https://prd-p-h9h24.splunkcloud.com/saml/acs

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

☑ Logout Consumer Service URL https://prd-p-h9h24.splunkcloud.com/saml/logout

If set, the user is redirected to the URL after successful logout.

Download the WebDM CA because you will need it later :



WebADM Server Administration

WebADM v2.3.13 (64bit) running on server Deb11-WebADM (nodeId: 5e30d4d8) in Standalone Mode.

Server Version Details: WebADM/2.3.13 Apache/2.4.58 PHP/8.1.27 OpenSSL/3.1.5
Internal Server Time: 2024-02-14 15:15:19 Etc/GMT-1 (**NTP check Ok**)
WebADM Features: WebApps (**Enabled**), WebSrvs (**Enabled**), Manager (**Enabled**)
RCDevs Cloud Services: BASE, LICENSE, SUPPORT, PUSH, SMS, PROOF (**Connected**)

Active LDAP Server: *LDAP Server (192.168.4.11)*
Active Session Server: *Session Server 1 (::1)*
Active SQL Server: *SQL Server 1 (127.0.0.1)*
Active PKI Server: *PKI Server 1 (127.0.0.1)*

User Domains (2)
Associate domain names with LDAP user search bases.

Client Policies (1)
Define custom policy settings for consumer applications.

Hosted Tenants (0)
Hosted customer tenant with dedicated LDAP subtree.

LDAP Option Sets (0)
LDAP subtree customizations, alerts and badging features.

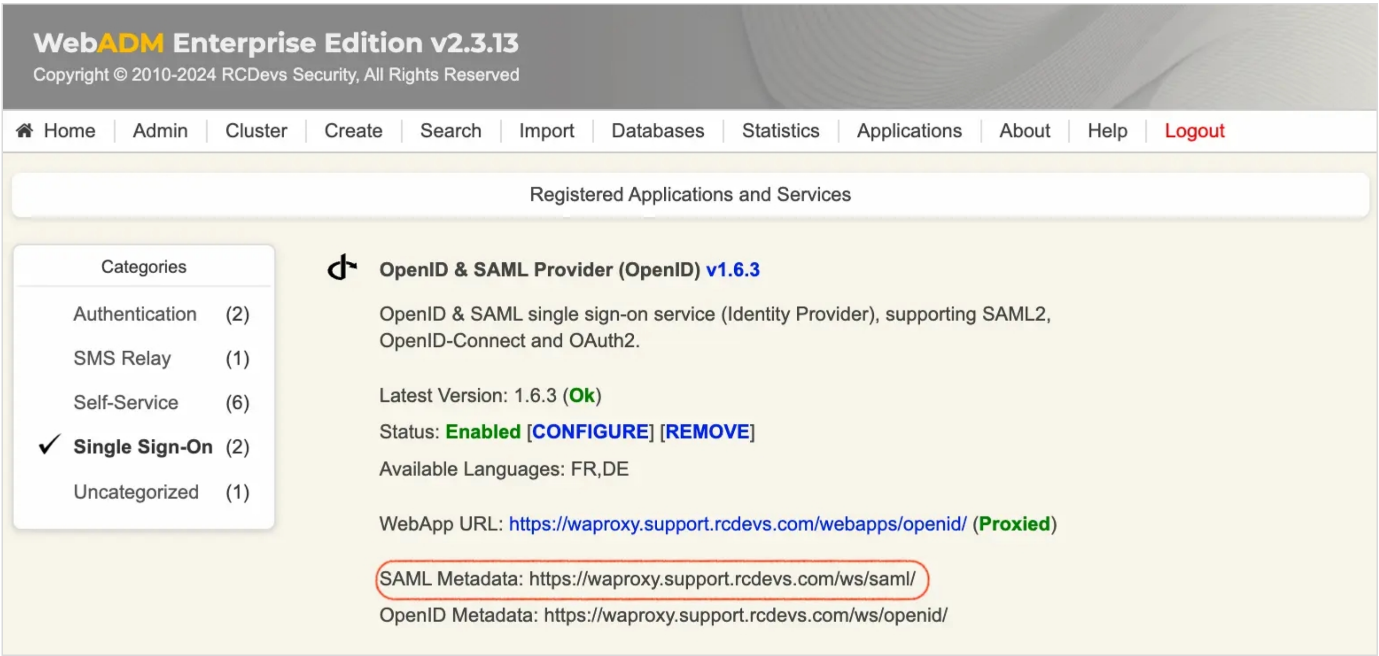
Administrator Roles (0)
Create admin role templates for your 'other' administrators.

LDAP Mount Points (0)
Connect secondary LDAP servers to the tree view.

Licensing and Configurations
- Software License Details
- LDAP Server Details
- LDAP Server Schema
- Memory Usage Details
- Hardware Modules Details
- Remote Manager Interface
- Config Object Statuses
- Network Service Statuses
- WebADM Base Settings
- Trusted CA Certificates

Runtime Actions
- Download Internal CA Certificate
- Create Server or Client Certificate
- Create Web Service API Keys
- Clear System & Application Caches (413 KB)
- Clear Application Sessions & Work Data
- Start Scheduled Background Tasks
- Reload WebADM Configurations

Here you can retrieve the SAML metadata of the IDP :


WebADM Enterprise Edition v2.3.13
Copyright © 2010-2024 RCDevs Security, All Rights Reserved
Home
Admin
Cluster
Create
Search
Import
Databases
Statistics
Applications
About
Help
Logout

Registered Applications and Services

Categories
- Authentication (2)
- SMS Relay (1)
- Self-Service (6)
- ✓ **Single Sign-On** (2)
- Uncategorized (1)

OpenID & SAML Provider (OpenID) v1.6.3

OpenID & SAML single sign-on service (Identity Provider), supporting SAML2, OpenID-Connect and OAuth2.

Latest Version: 1.6.3 (**Ok**)
Status: **Enabled** [**CONFIGURE**] [**REMOVE**]
Available Languages: FR,DE

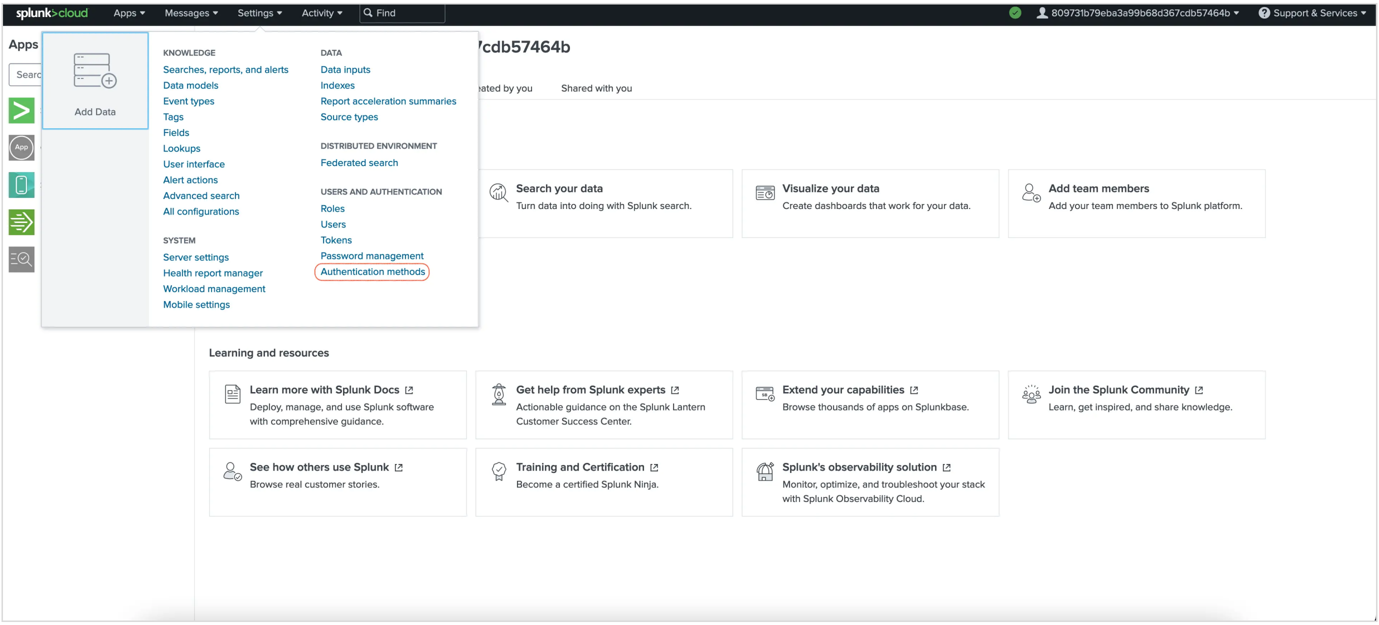
WebApp URL: https://waproxy.support.rcdevs.com/webapps/openid/ (**Proxied**)

SAML Metadata: https://waproxy.support.rcdevs.com/ws/saml/
OpenID Metadata: https://waproxy.support.rcdevs.com/ws/openid/

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="waproxy.support.rcdevs.com">
<IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIFJDCCAwygAwIBAgIRAJ6ZaPKBwLhG+K3PmGqkGygwDQYJKoZIhvcNAQELBQAwUjEaMBg

<!--  Cert Fingerprint (SHA1): f15dfe8d61c2e4f340c158bd5b30b739c668debd  -->
<!--  Cert Fingerprint (SHA256):
37c9adedbe69baa2237b6c822e7d8ca930eded9dfc2ef532c06780a7950cbe8e  -->
<!--  Cert Fingerprint (MD5): 9c0e456cdee22ef17f62eec4c0155341  -->
</X509Data>
</KeyInfo>
</KeyDescriptor>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://waproxy.support.rcdevs.com/openid/index.php"/>
</IDPSSODescriptor>
</EntityDescriptor>
```

Now it's time to set up SAML on Splunk Cloud. In the dashboard, click on `Settings`, then select `Authentication Methods`.



Search your data
Turn data into doing with Splunk search.

Visualize your data
Create dashboards that work for your data.

Add team members
Add your team members to Splunk platform.

Learn more with Splunk Docs
Deploy, manage, and use Splunk software with comprehensive guidance.

Get help from Splunk experts
Actionable guidance on the Splunk Lantern Customer Success Center.

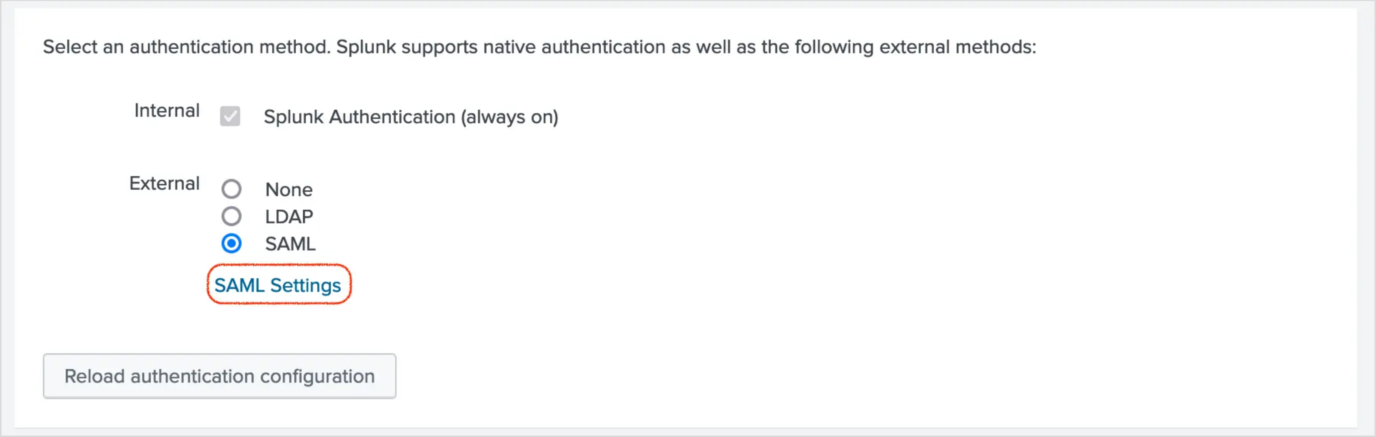
Extend your capabilities
Browse thousands of apps on Splunkbase.

Join the Splunk Community
Learn, get inspired, and share knowledge.

See how others use Splunk
Browse real customer stories.

Training and Certification
Become a certified Splunk Ninja.

Splunk's observability solution
Monitor, optimize, and troubleshoot your stack with Splunk Observability Cloud.

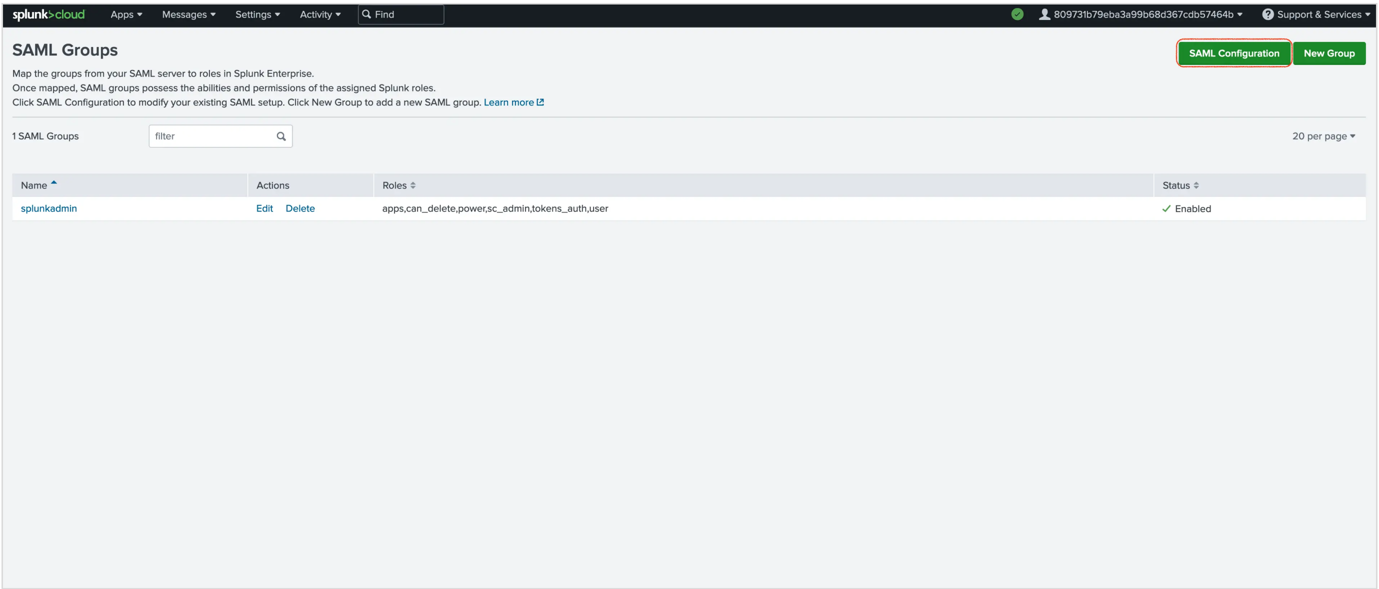


Select an authentication method. Splunk supports native authentication as well as the following external methods:

Internal ☑ Splunk Authentication (always on)

External ○ None
○ LDAP
● SAML

SAML Configuration :



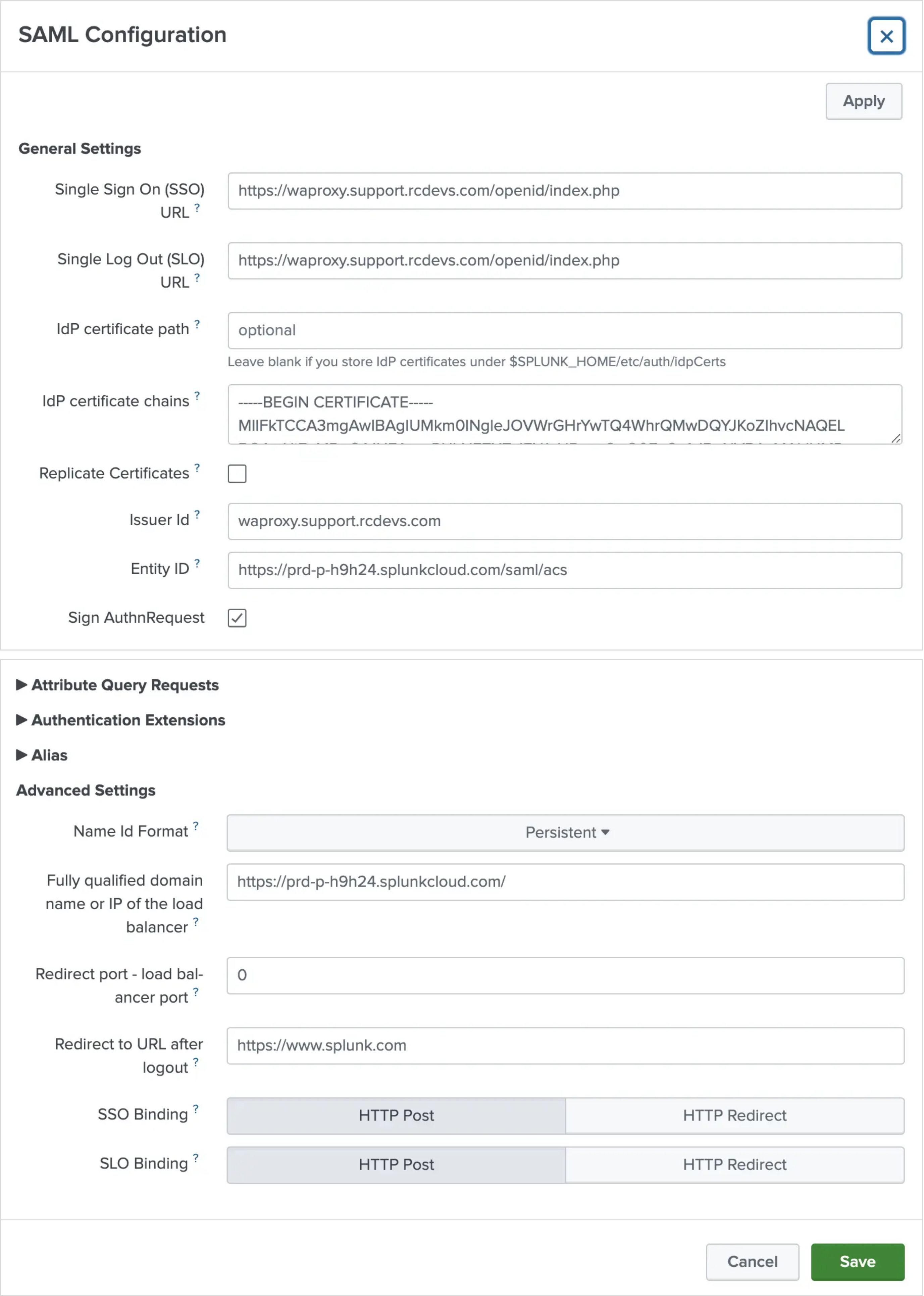
SAML Groups

Map the groups from your SAML server to roles in Splunk Enterprise.
Once mapped, SAML groups possess the abilities and permissions of the assigned Splunk roles.
Click SAML Configuration to modify your existing SAML setup. Click New Group to add a new SAML group. Learn more

1 SAML Groups filter 20 per page

Name	Actions	Roles	Status
splunkadmin	Edit Delete	apps,can_delete,power,sc_admin,tokens_auth,user	✓ Enabled

Upload the IDP metadata into Metadata Contents to obtain the following configurations



SAML Configuration

General Settings

Single Sign On (SSO) URL ?: https://waproxy.support.rcdevs.com/openid/index.php

Single Log Out (SLO) URL ?: https://waproxy.support.rcdevs.com/openid/index.php

IdP certificate path ?: optional

Leave blank if you store IdP certificates under $SPLUNK_HOME/etc/auth/idpCerts

IdP certificate chains ?:

```
-----BEGIN CERTIFICATE-----
MIIFkTCCA3mgAwIBAgIUMkm0INgleJOVWrGHrYwTQ4WhrQMwDQYJKoZIhvcNAQEL
```

Replicate Certificates ? □

Issuer Id ?: waproxy.support.rcdevs.com

Entity ID ?: https://prd-p-h9h24.splunkcloud.com/saml/acs

Sign AuthnRequest ☑

▶ **Attribute Query Requests**

▶ **Authentication Extensions**

▶ **Alias**

Advanced Settings

Name Id Format ?: Persistent ▾

Fully qualified domain name or IP of the load balancer ?: https://prd-p-h9h24.splunkcloud.com/

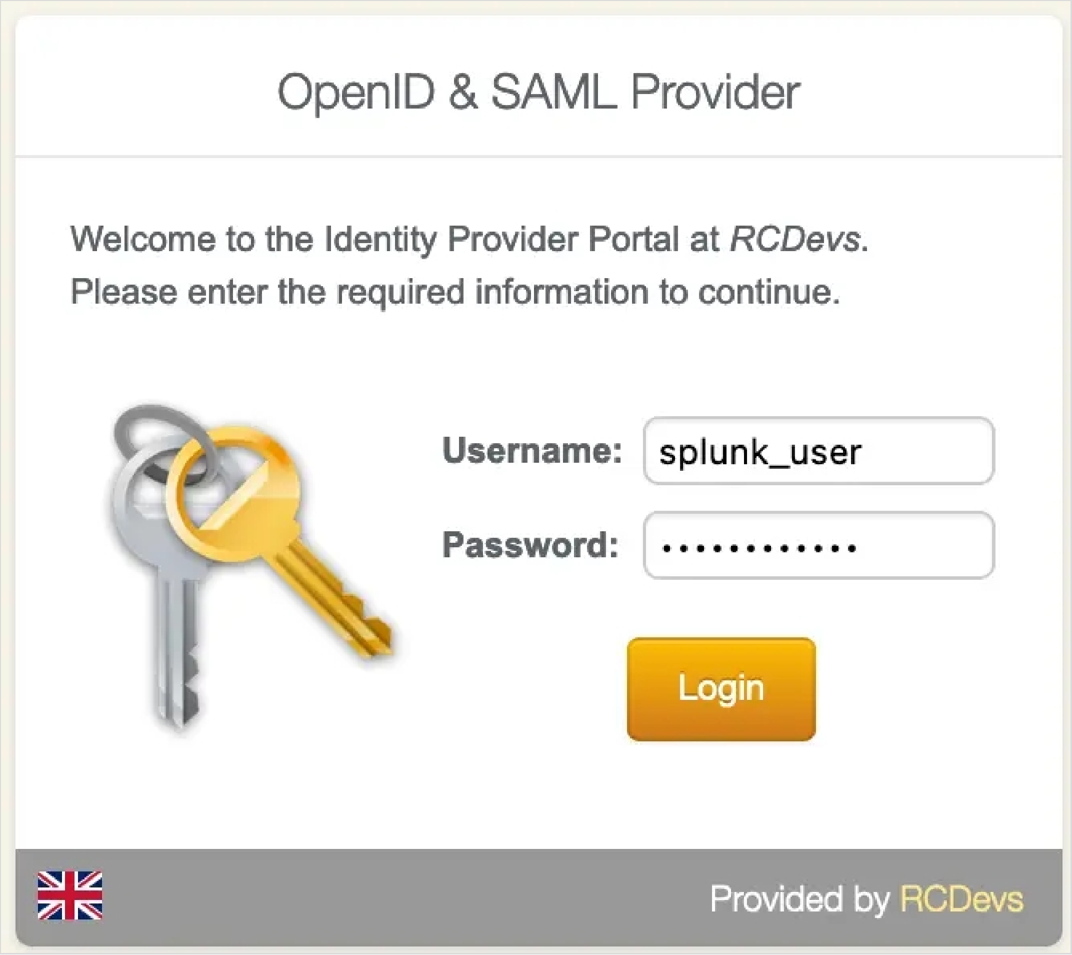
Redirect port - load balancer port ?: 0

Redirect to URL after logout ?: https://www.splunk.com

SSO Binding ?: [✓ HTTP Post] HTTP Redirect

SLO Binding ?: [✓ HTTP Post] HTTP Redirect

Open the SP link in a private browser and log in with your user account :

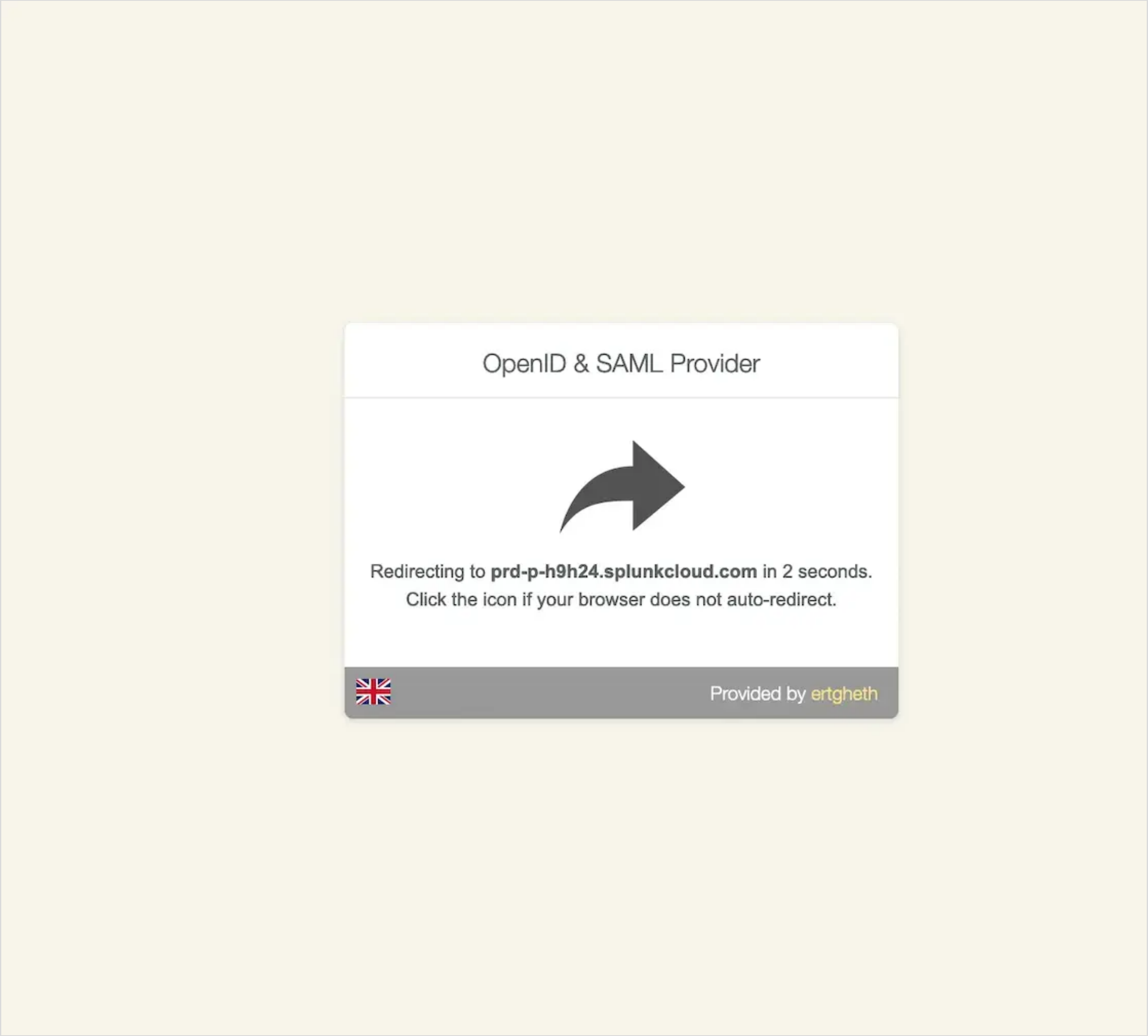


OpenID & SAML Provider

Welcome to the Identity Provider Portal at *RCDevs*.
Please enter the required information to continue.

Username: splunk_user

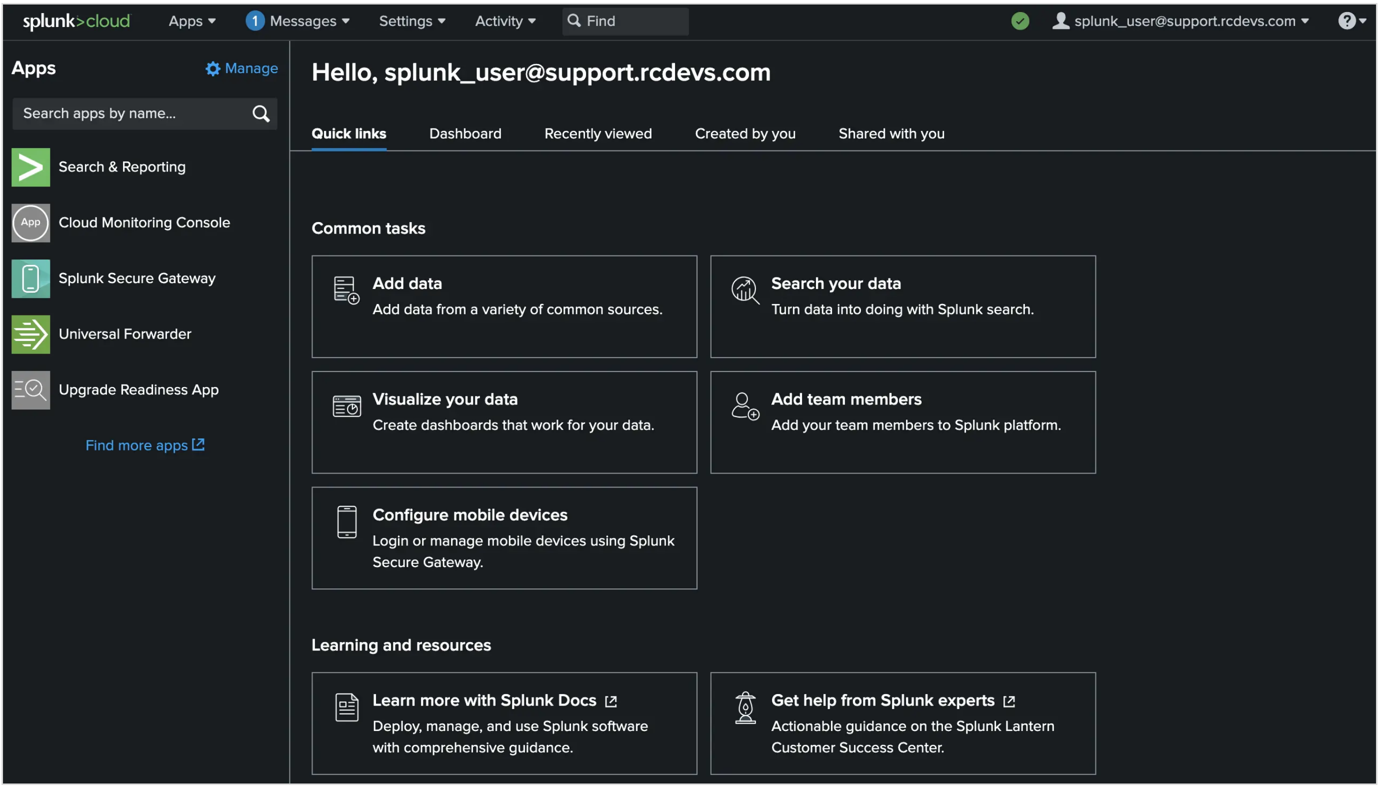
Password: •••••••••••••



OpenID & SAML Provider

Redirecting to **prd-p-h9h24.splunkcloud.com** in 2 seconds.
Click the icon if your browser does not auto-redirect.

Provided by ertgheth



Hello, splunk_user@support.rcdevs.com

Common tasks

Add data
Add data from a variety of common sources.

Search your data
Turn data into doing with Splunk search.

Visualize your data
Create dashboards that work for your data.

Add team members
Add your team members to Splunk platform.

Configure mobile devices
Login or manage mobile devices using Splunk Secure Gateway.

Learning and resources

Learn more with Splunk Docs
Deploy, manage, and use Splunk software with comprehensive guidance.

Get help from Splunk experts
Actionable guidance on the Splunk Lantern Customer Success Center.

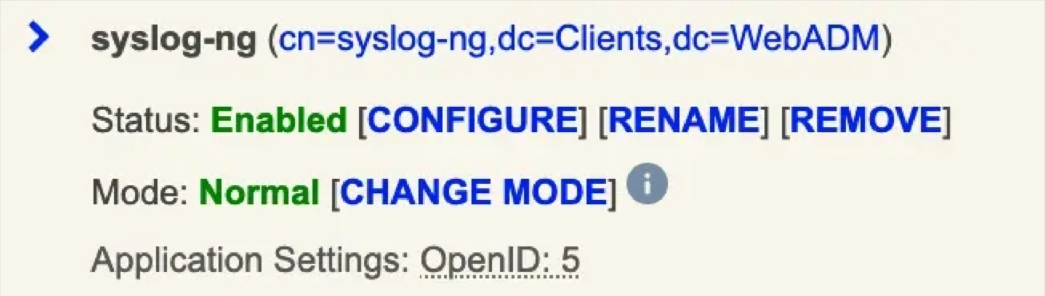
3.3.13 Syslog-ng store box (OpenID)

⚠ Note

For this integration, I used a local user that I created in syslog-ng with the necessary permissions. This user also exists in my WebADM. Alternatively, there is the option to use Active Directory as an LDAP backend. To do this, I recommend referring to the syslog-ng Store Box documentation.

To use WebADM as an IDP for Syslog-ng STORE BOX via OpenID, you will need :

Configure a client policies :

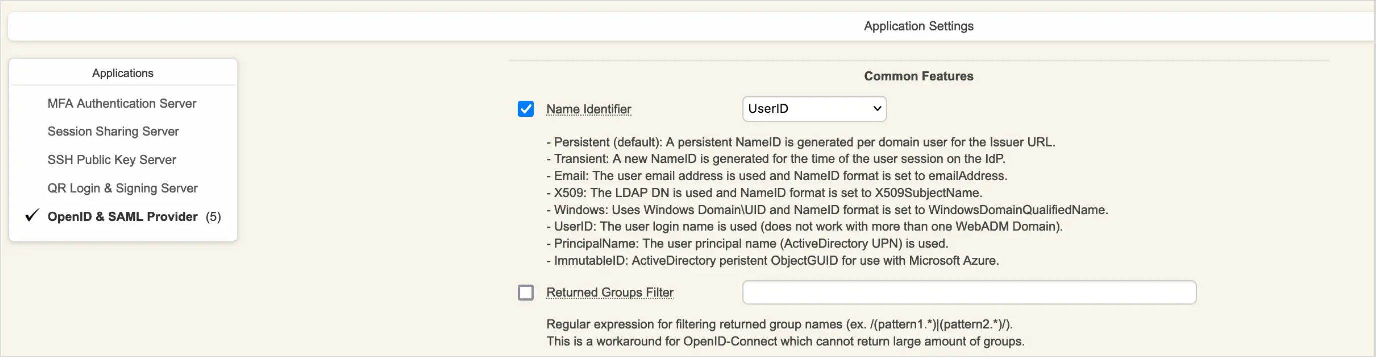


syslog-ng (cn=syslog-ng,dc=Clients,dc=WebADM)

Status: **Enabled** [**CONFIGURE**] [**RENAME**] [**REMOVE**]

Mode: **Normal** [**CHANGE MODE**]

Application Settings: OpenID: 5



Application Settings

Applications

- MFA Authentication Server
- Session Sharing Server
- SSH Public Key Server
- QR Login & Signing Server
- ✓ **OpenID & SAML Provider** (5)

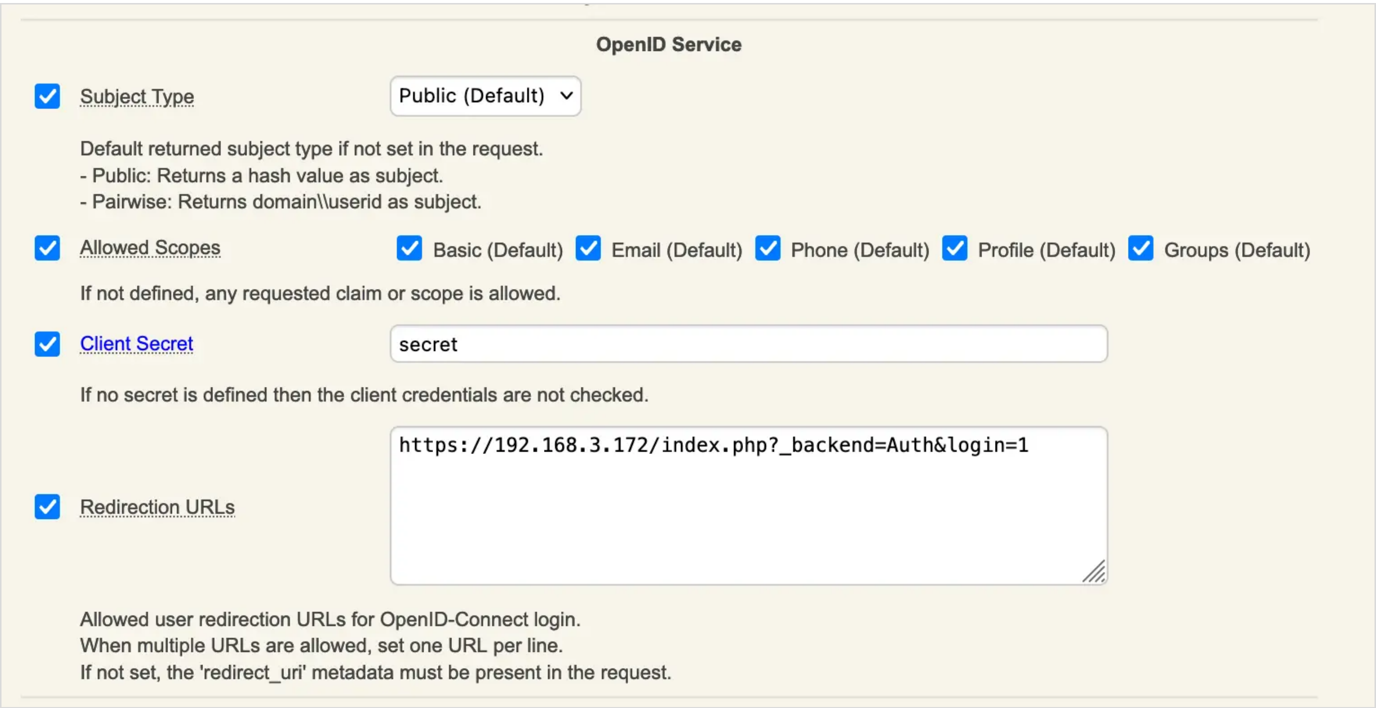
Common Features

☑ Name Identifier: UserID

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).
- PrincipalName: The user principal name (ActiveDirectory UPN) is used.
- ImmutableID: ActiveDirectory peristent ObjectGUID for use with Microsoft Azure.

☐ Returned Groups Filter

Regular expression for filtering returned group names (ex. /(pattern1.*)|(pattern2.*)/).
This is a workaround for OpenID-Connect which cannot return large amount of groups.



OpenID Service

☑ Subject Type: Public (Default)

Default returned subject type if not set in the request.
- Public: Returns a hash value as subject.
- Pairwise: Returns domain\\userid as subject.

☑ Allowed Scopes: ☑ Basic (Default) ☑ Email (Default) ☑ Phone (Default) ☑ Profile (Default) ☑ Groups (Default)

If not defined, any requested claim or scope is allowed.

☑ Client Secret: secret

If no secret is defined then the client credentials are not checked.

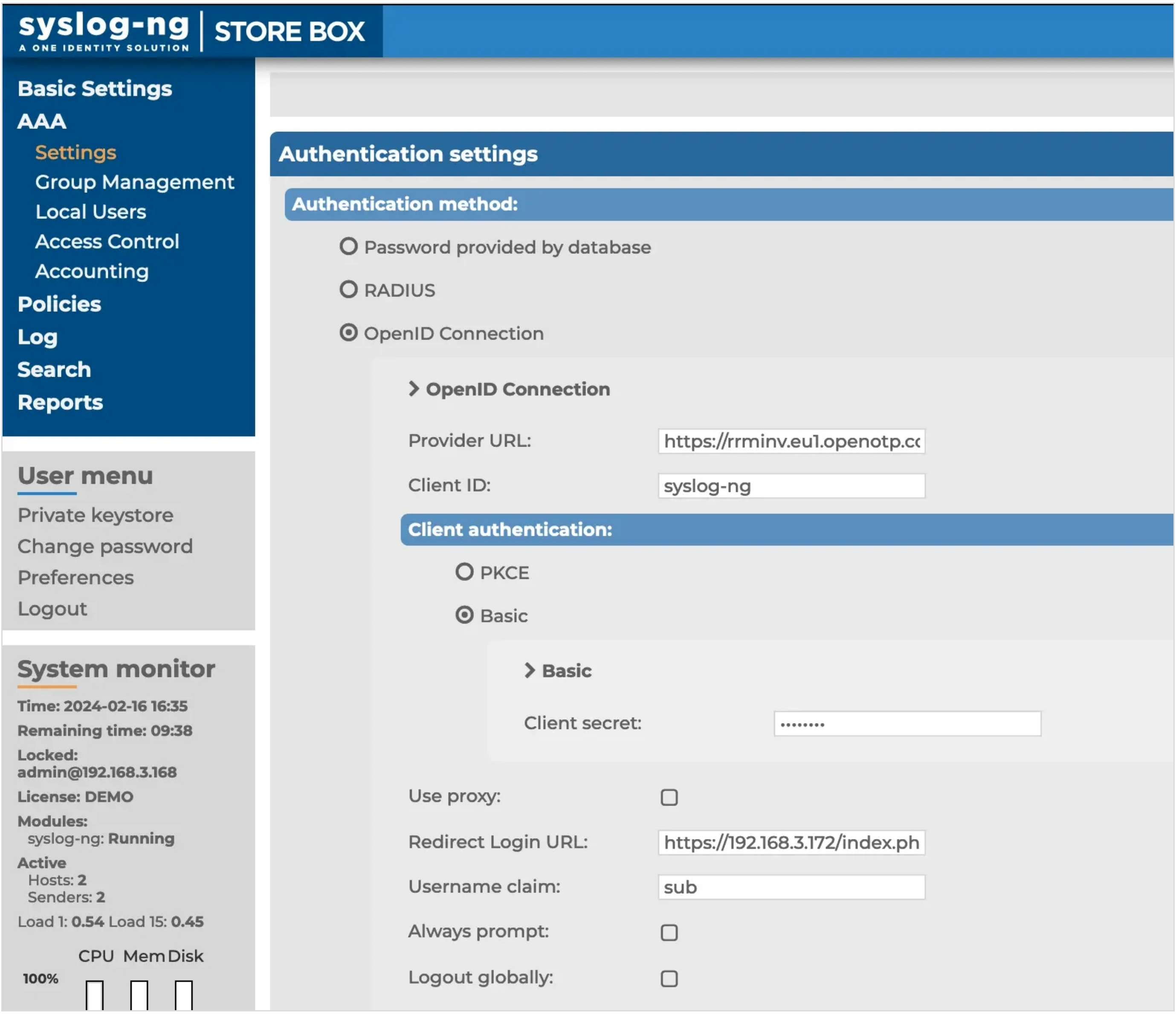
☑ Redirection URLs:

```
https://192.168.3.172/index.php?_backend=Auth&login=1
```

Allowed user redirection URLs for OpenID-Connect login.
When multiple URLs are allowed, set one URL per line.
If not set, the 'redirect_uri' metadata must be present in the request.

`Redirection URLs` can be found in the default settings of the Service Provider under the section `Redirect Login URL`

And now we will configure Syslog-ng Store Box :



Time: 2024-02-16 16:35
Remaining time: 09:38
Locked: admin@192.168.3.168
License: DEMO
Modules:
syslog-ng: Running
Active
Hosts: 2
Senders: 2
Load 1: 0.54 Load 15: 0.45

Authentication settings

Authentication method:

○ Password provided by database

○ RADIUS

● OpenID Connection

> OpenID Connection

Provider URL: https://rrminv.eu1.openotp.cc

Client ID: syslog-ng

Client authentication:

○ PKCE

● Basic

> Basic

Client secret: ••••••••

Use proxy: □

Redirect Login URL: https://192.168.3.172/index.ph

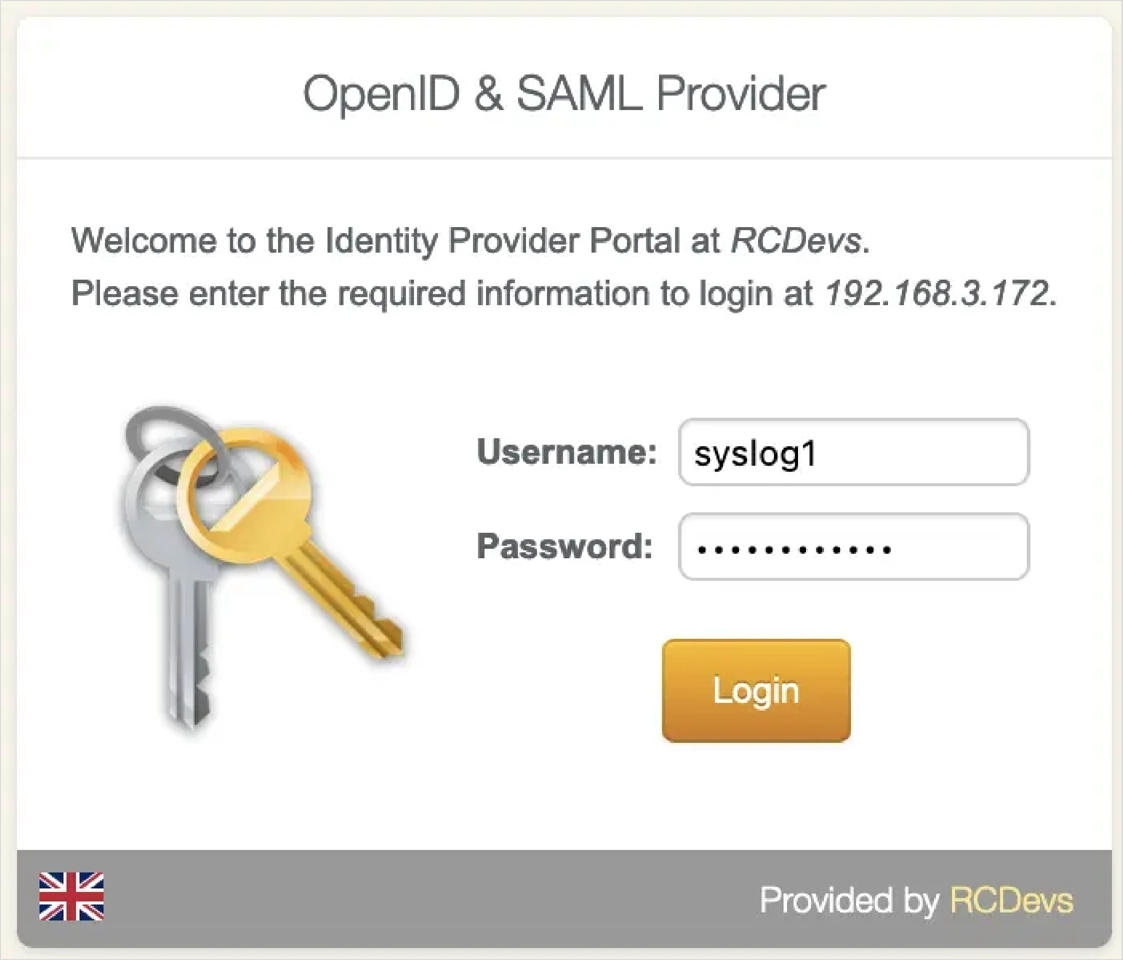
Username claim: sub

Always prompt: □

Logout globally: □

The `Provider URL` is the WebApp URL of OpenID. And the `Client secret` is the one configured in our client policies

Test login :

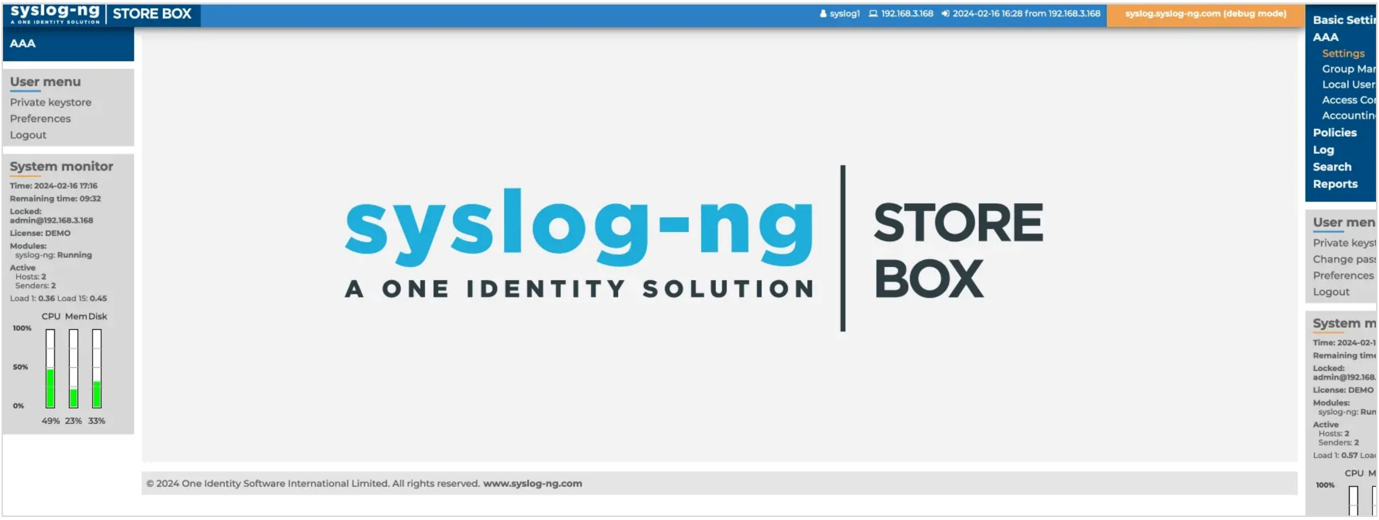


OpenID & SAML Provider

Welcome to the Identity Provider Portal at *RCDevs*.
Please enter the required information to login at *192.168.3.172*.

Username: syslog1

Password: •••••••••••••



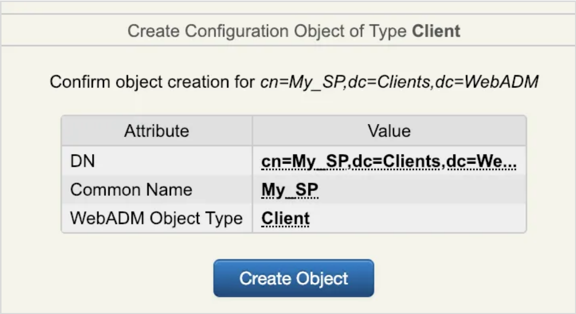
4. How to Create and match a client policy per Service Provider

Since the WebADM 1.6.9-x and OpenID/SAML provider 1.3.0, it is possible to create WebADM client policies per Service Provider. That will allow you to return attributes, nameID, attributes mappings, or use a different certificate per client (SP) and not only globally. This feature makes the IDP much more powerful and provide flexibility for each client integrations.

4.1 SP Initiated mode

To create a client policy for your SP in SP initiated mode, log in on the WebADM Admin GUI, click on `Admin` tab, `Client Policy` and click on `Add Client`.

Give a name to your Client Policy and then click `Proceed` and `Create Object`.

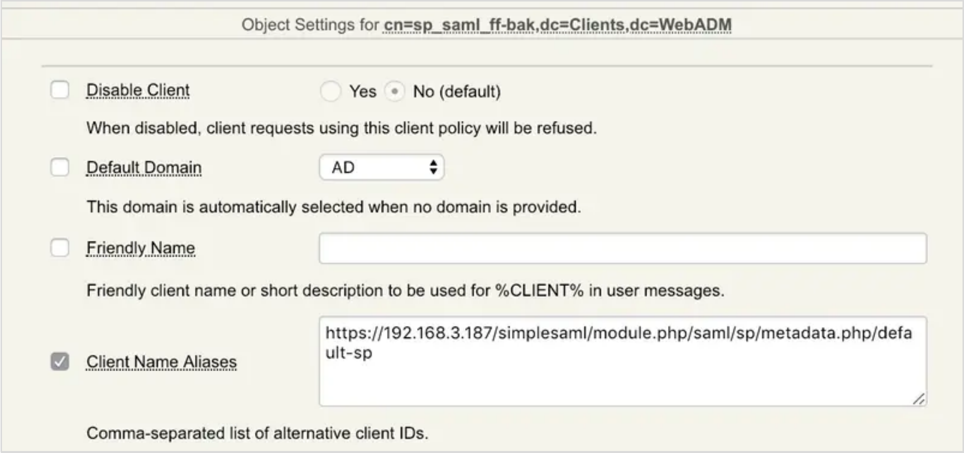


Create Configuration Object of Type **Client**

Confirm object creation for *cn=My_SP,dc=Clients,dc=WebADM*

Attribute	Value
DN	cn=My_SP,dc=Clients,dc=We...
Common Name	My_SP
WebADM Object Type	Client

We will now configure the client policy. Many settings can be applied here like which users/groups/networks the client policy will be applied, allowed/excluded hours, which domain… An important setting on this page is the Client Name Aliases which will allow us to do the matching between the client policy and the SP. For this, the client policy must be created with the SP issuer URL (Entity ID) as Client Name Aliases.



Object Settings for cn=sp_saml_ff-bak,dc=Clients,dc=WebADM

☐ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☐ Default Domain AD

This domain is automatically selected when no domain is provided.

☐ Friendly Name

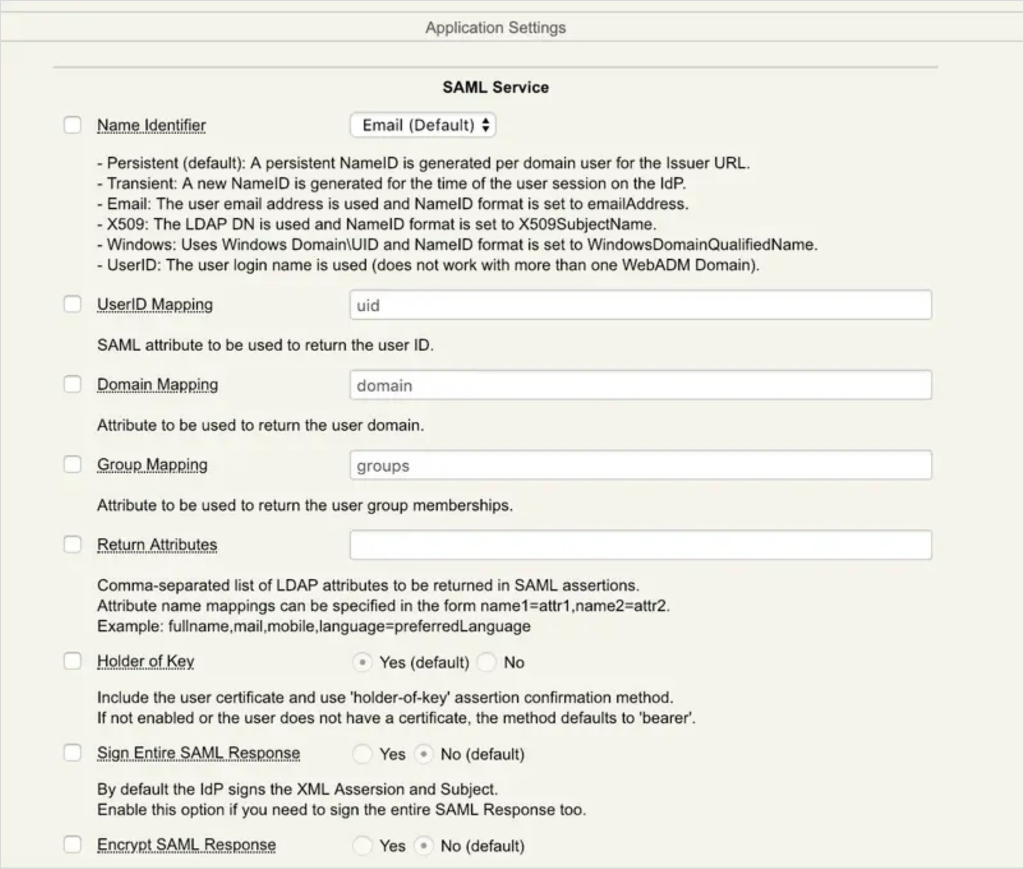
Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases https://192.168.3.187/simplesaml/module.php/saml/sp/metadata.php/default-sp

Comma-separated list of alternative client IDs.

The matching is done, we will now configure the SP policy.

If you scroll down a little bit, you will find the setting named `Forced Application Policies`, click on the `Edit` button and select `OpenID` application in the left box.



Application Settings

SAML Service

□ Name Identifier — Email (Default)

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).

□ UserID Mapping — uid

SAML attribute to be used to return the user ID.

□ Domain Mapping — domain

Attribute to be used to return the user domain.

□ Group Mapping — groups

Attribute to be used to return the user group memberships.

□ Return Attributes

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

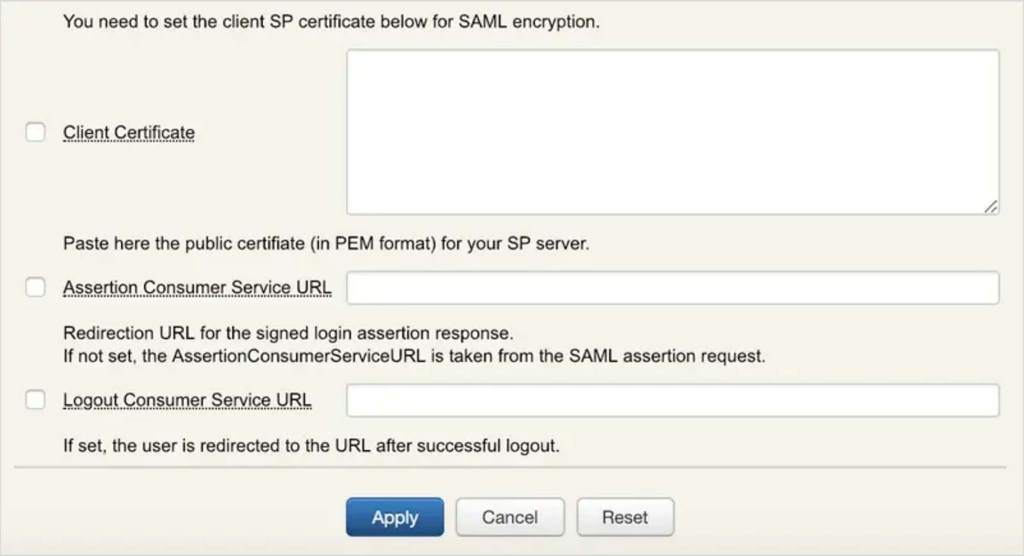
□ Holder of Key — ● Yes (default) ○ No

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

□ Sign Entire SAML Response — ○ Yes ● No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

□ Encrypt SAML Response — ○ Yes ● No (default)



You need to set the client SP certificate below for SAML encryption.

□ Client Certificate

Paste here the public certifiate (in PEM format) for your SP server.

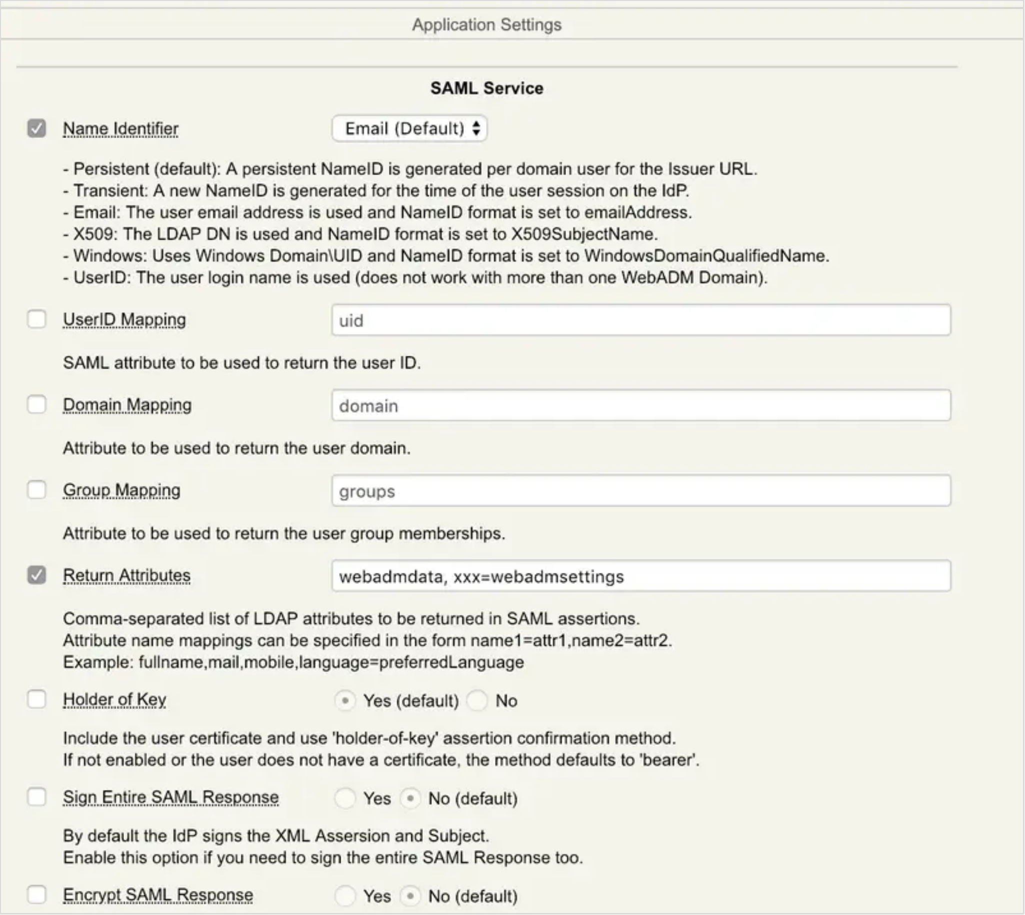
□ Assertion Consumer Service URL

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

□ Logout Consumer Service URL

If set, the user is redirected to the URL after successful logout.

Configure your client policy with every setting you need for your SP and then save your configuration.



Application Settings

SAML Service

☑ Name Identifier — Email (Default)

- Persistent (default): A persistent NameID is generated per domain user for the Issuer URL.
- Transient: A new NameID is generated for the time of the user session on the IdP.
- Email: The user email address is used and NameID format is set to emailAddress.
- X509: The LDAP DN is used and NameID format is set to X509SubjectName.
- Windows: Uses Windows Domain\UID and NameID format is set to WindowsDomainQualifiedName.
- UserID: The user login name is used (does not work with more than one WebADM Domain).

□ UserID Mapping — uid

SAML attribute to be used to return the user ID.

□ Domain Mapping — domain

Attribute to be used to return the user domain.

□ Group Mapping — groups

Attribute to be used to return the user group memberships.

☑ Return Attributes — webadmdata, xxx=webadmsettings

Comma-separated list of LDAP attributes to be returned in SAML assertions.
Attribute name mappings can be specified in the form name1=attr1,name2=attr2.
Example: fullname,mail,mobile,language=preferredLanguage

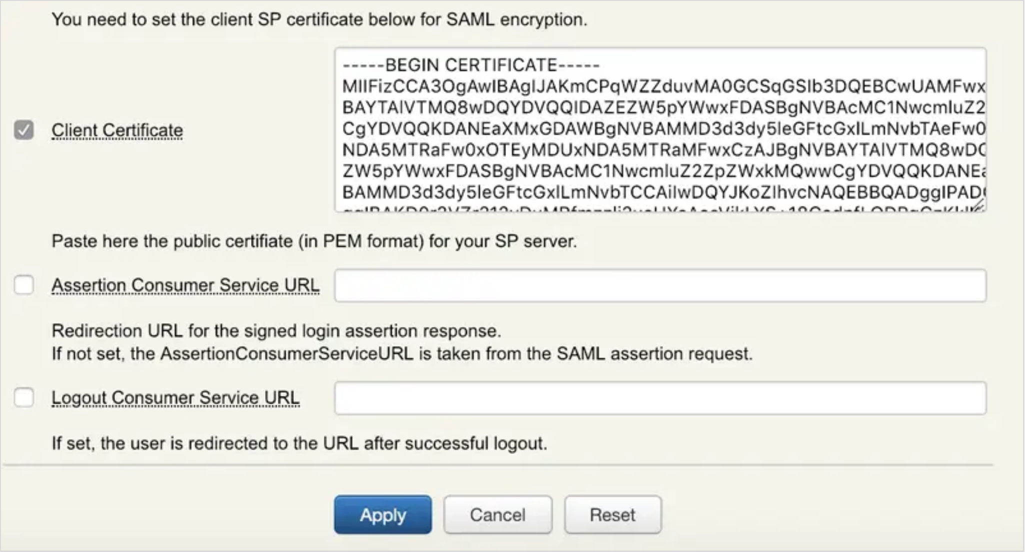
□ Holder of Key — ● Yes (default) ○ No

Include the user certificate and use 'holder-of-key' assertion confirmation method.
If not enabled or the user does not have a certificate, the method defaults to 'bearer'.

□ Sign Entire SAML Response — ○ Yes ● No (default)

By default the IdP signs the XML Assersion and Subject.
Enable this option if you need to sign the entire SAML Response too.

□ Encrypt SAML Response — ○ Yes ● No (default)



You need to set the client SP certificate below for SAML encryption.

☑ Client Certificate

```
-----BEGIN CERTIFICATE-----
MIIFizCCA3OgAwIBAgIJAKmCPqWZZduvMA0GCSqGSIb3DQEBCwUAMFwx
BAYTAlVTMQ8wDQYDVQQIDAZEZW5pYWwxFDASBgNVBAcMC1NwcmluZ2
CgYDVQQKDANEaXMxGDAWBgNVBAMMD3d3dy5leGFtcGxlLmNvbTAeFw0
NDA5MTRaFw0xOTEyMDUxNDA5MTRaMFwxCzAJBgNVBAYTAlVTMQ8wD
ZW5pYWwxFDASBgNVBAcMC1NwcmluZ2ZpZWxkMQwwCgYDVQQKDANE
BAMMD3d3dy5leGFtcGxlLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPAD
```

Paste here the public certifiate (in PEM format) for your SP server.

□ Assertion Consumer Service URL

Redirection URL for the signed login assertion response.
If not set, the AssertionConsumerServiceURL is taken from the SAML assertion request.

□ Logout Consumer Service URL

If set, the user is redirected to the URL after successful logout.

Your client policy for your SP is now configured. Try an authentication from your SP and check the WebADM logs to be sure that your policy is applied correctly.

Note

You can not yet apply any OpenOTP settings in the same OpenID/SAML client policy. That part is in the RCDevs roadmap and will be added in the future.

4.2 IDP initiated mode

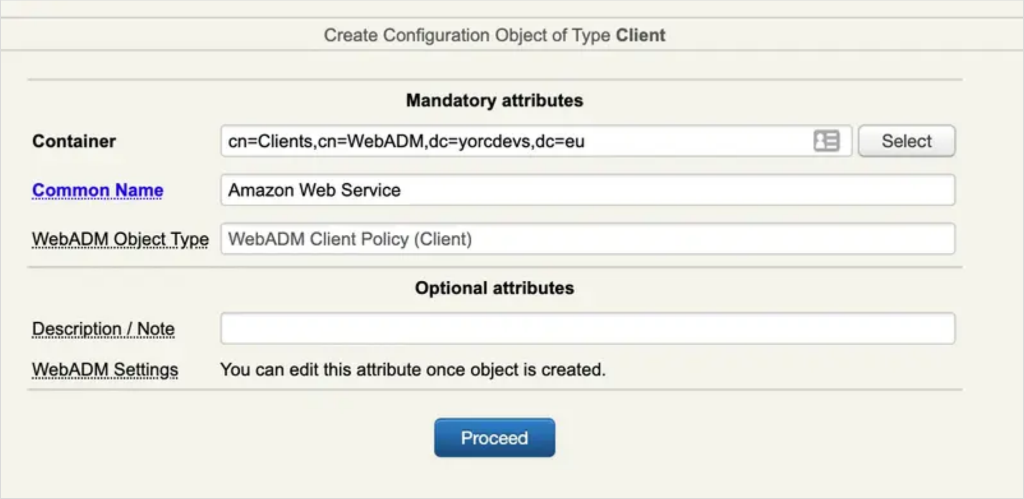
The way to create a client policy in IDP initiated mode is similar to SP initiated mode. The matching is done through the issuer value configured in the `app.ini` file located in `/opt/webadm/webapps/openid/apps/<application>.ini`

E.g for Amazon

```
[root@webadm1 ~]# cat opt/webadm/webapps/openid/apps/amazonws.ini

name   = "Amazon WS"
help   = "Amazon Web Services (AWS)"
method = "HTTP-POST"
source = "https://signin.aws.amazon.com/saml"
issuer = "https://signin.aws.amazon.com"
nameid = "Persistent"
```

I can then create my policy for AWS like below :



Create Configuration Object of Type **Client**

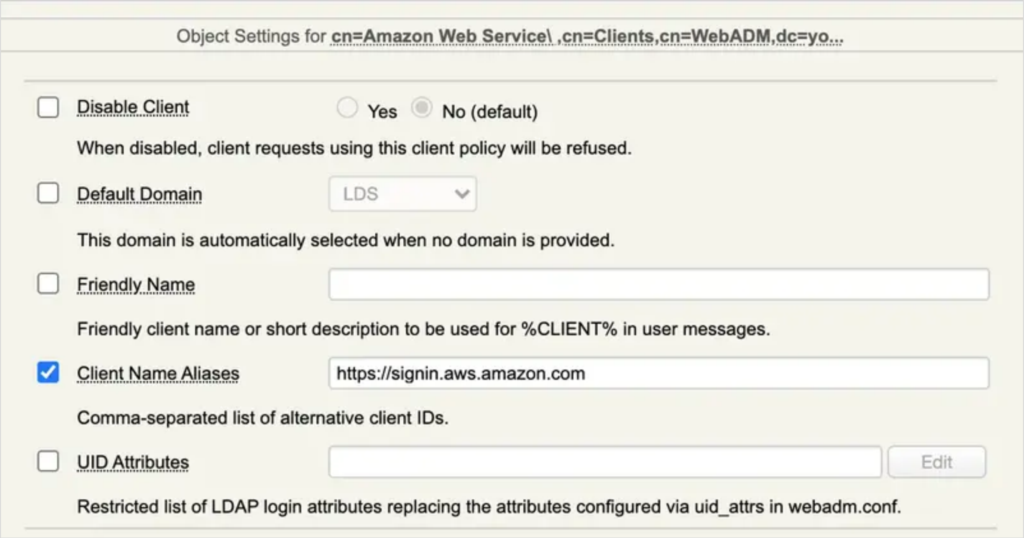
Mandatory attributes

Container	cn=Clients,cn=WebADM,dc=yorcdevs,dc=eu
Common Name	Amazon Web Service
WebADM Object Type	WebADM Client Policy (Client)

Optional attributes

Description / Note	
WebADM Settings	You can edit this attribute once object is created.

After creating the client policy object, I configure the client name alias for the matching operate :



Object Settings for cn=Amazon Web Service\ ,cn=Clients,cn=WebADM,dc=yo...

☐ Disable Client ○ Yes ● No (default)

When disabled, client requests using this client policy will be refused.

☐ Default Domain LDS

This domain is automatically selected when no domain is provided.

☐ Friendly Name

Friendly client name or short description to be used for %CLIENT% in user messages.

☑ Client Name Aliases https://signin.aws.amazon.com

Comma-separated list of alternative client IDs.

☐ UID Attributes

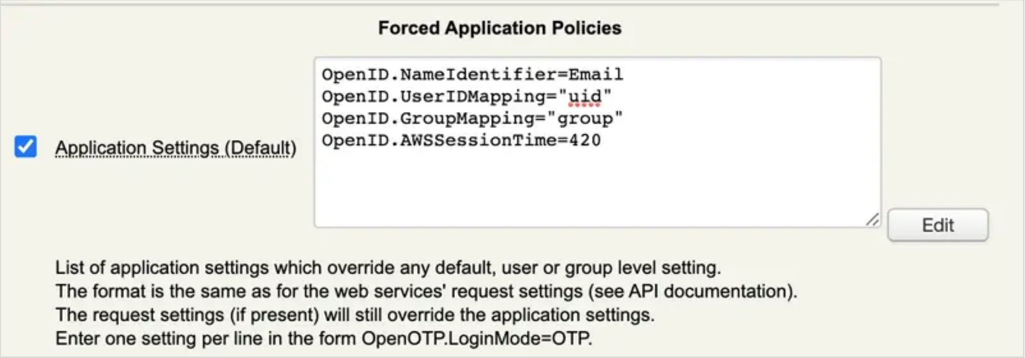
Restricted list of LDAP login attributes replacing the attributes configured via uid_attrs in webadm.conf.

In the next section, we show you how to return attributes for AWS SP.

4.3 Returned attributes and attribut mapping

4.3.1 General attributs

Here, I configured some returned attribute to be returned to AWS :



Forced Application Policies

☑ Application Settings (Default)

```
OpenID.NameIdentifier=Email
OpenID.UserIDMapping="uid"
OpenID.GroupMapping="group"
OpenID.AWSSessionTime=420
```

List of application settings which override any default, user or group level setting.
The format is the same as for the web services' request settings (see API documentation).
The request settings (if present) will still override the application settings.
Enter one setting per line in the form OpenOTP.LoginMode=OTP.

Note

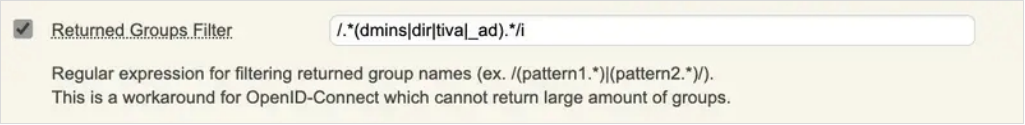
You can not yet apply any OpenOTP settings in the same OpenID/SAML client policy. That part is in the RCDevs roadmap and will be added in the future.

4.3.2 Group filtering in SAML/OpenID responses

In the general configuration of SAML/OpenID or on a per-SP (Service Provider) client policy basis, you have the option to limit the groups that are included in the SAML assertion or OpenID response. This feature proves especially valuable with OpenID, particularly when users belong to a large number of groups. In such cases, including all these groups in the JWT (JSON Web Token) can lead to issues, such as exceeding the maximum size of HTTP headers.

To address this limitation, RCDevs has implemented a solution that allows you to define regular expressions (regex) to filter and include only those groups that match the specified regex pattern. Below, you will find a few examples of regex expressions:

```
\b(?:domain|direct*)\b
\b(?:domain|dir.*)\b
/(.*dir*.)|(domain.*)/
/\b(super_admin|Schema Admins|Indirect2|activated)\b/i
/.*(dmins|dir|tival_ad).*/i
```



☑ Returned Groups Filter: /.*(dmins|dir|tival_ad).*/i

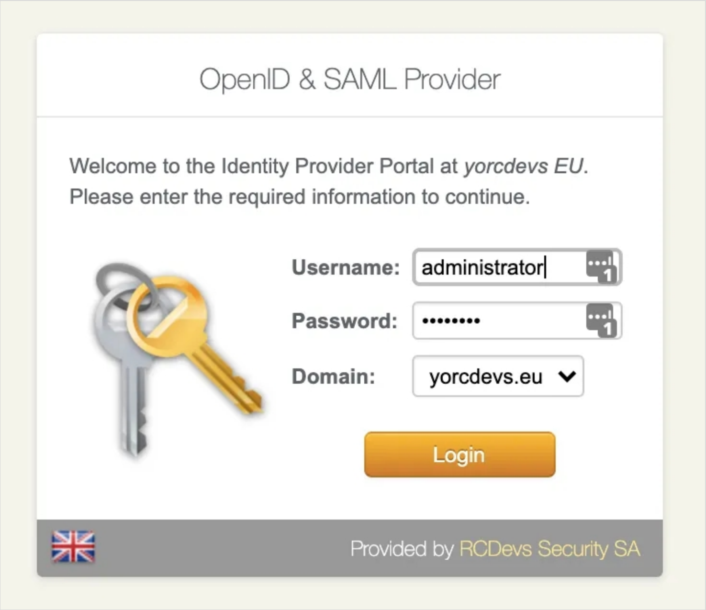
Regular expression for filtering returned group names (ex. /(pattern1.*)|(pattern2.*)/).
This is a workaround for OpenID-Connect which cannot return large amount of groups.

The /i option in the regex makes the pattern matching case-insensitive. Here is what is returned when my regex expression is applied:

```
"groups": [
  "activated",
  "indirect2",
  "direct",
  "super_admin",
  "domain admins",
  "schema admins",
  "indirect"
]
```

4.4 Test login with AWS

My AWS service provider is now configured with my WebADM IDP. I can perform a login on OpenID & SAML Provider web application and access to AWS :



OpenID & SAML Provider

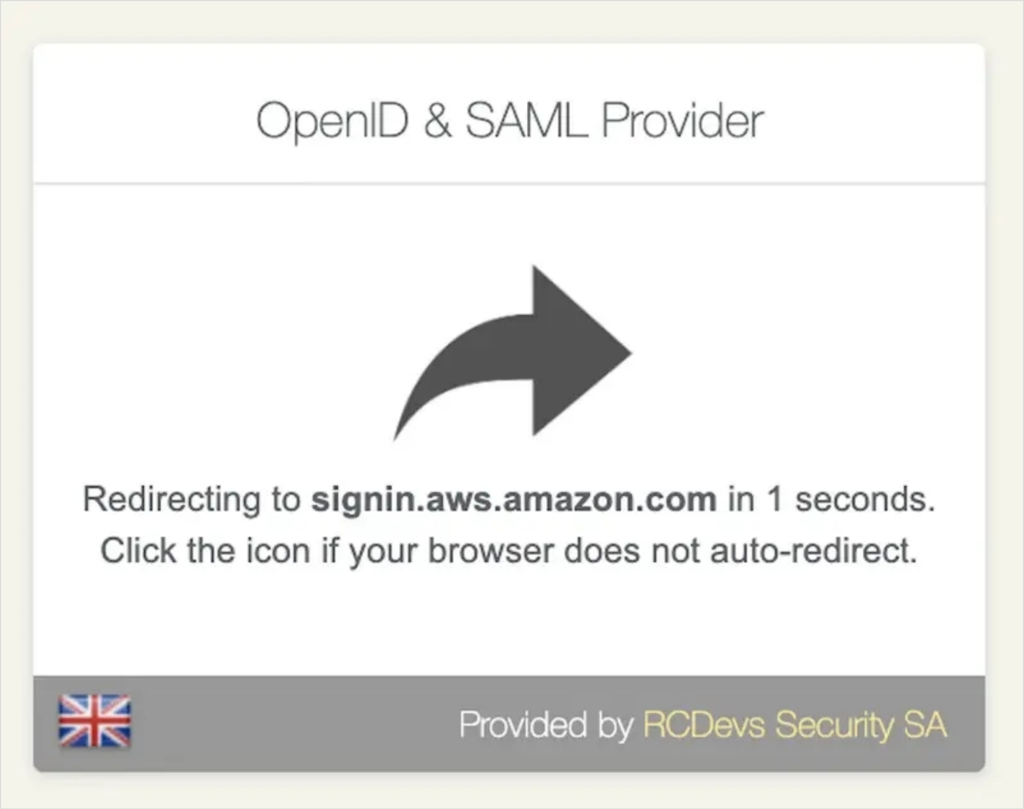
Welcome to the Identity Provider Portal at *yorcdevs EU*.
Please enter the required information to continue.

Username: administrator

Password: ••••••••

Domain: yorcdevs.eu

After a success login on the IDP, if no other SP are configured with your IDP, you are automatically redirected to AWS page :

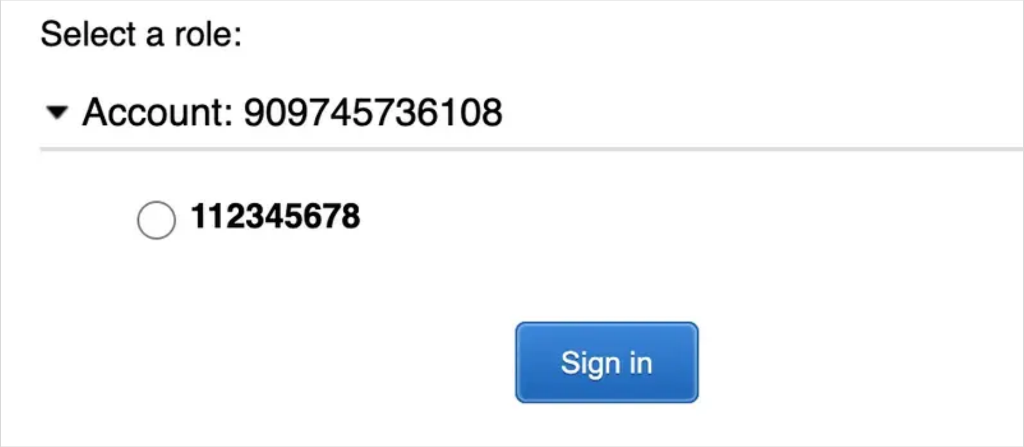


OpenID & SAML Provider

Redirecting to **signin.aws.amazon.com** in 1 seconds.
Click the icon if your browser does not auto-redirect.

Provided by RCDevs Security SA

After the redirection to AWS login page, you are prompted to select the role you want to use with your account. If multiple roles are configued under the user or group, then all role allowed by the user are returned and can be choosen by the end user :



Select a role:

▾ Account: 909745736108

○ **112345678**

Click `Sign In` button you are now connected to AWS with your account and the associated role.

5. Login debug

5.1 SAML request

To check your configued attributes are well returned by WebADM IDP in the SAML assertion, you can the browser extension SAML Message Decoder available on Chrome. Perform a login request and check the SAML Message Decoder console. You should see something similar :

```
<?xml version="1.0"?>
<samlp:Response Destination="https://signin.aws.amazon.com/saml"
```

```
    ID="_f8a62989fac5142a21d93c10fa6882e6f284b0314c" IssueInstant="2020-10-26T09:26:46Z"
Version="2.0"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
    <saml:Issuer>waproxy.support.rcdevs.com/</saml:Issuer>
    <samlp:Status><samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
    <saml:Assertion ID="_5490a6d31dd1a3c782a48d0ec1e1541b16756ac843" IssueInstant="2020-10-
26T09:26:46Z"
        Version="2.0">
        <saml:Issuer>https://waproxy.support.rcdevs.com/webapps/openid/</saml:Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:SignedInfo><ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
                <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                <ds:Reference URI="#_5490a6d31dd1a3c782a48d0ec1e1541b16756ac843">
                    <ds:Transforms><ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/><ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/></ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:DigestValue>qpLOfz9w9BlUANTvx7C7kB2DiImyIYHWjZYXNRvGPog=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>

<ds:SignatureValue>WncS2uxIpx2uKX4MmDlNAXWgjNBS4ZFfNZdFjrp6EXXBUnQkNblL1kCGNWPnCgsbR9pQ

            <ds:KeyInfo>
                <ds:X509Data>

<ds:X509Certificate>MIIDBjCCAe6gAwIBAgIBAjANBgkqhkiG9w0BAQsFADAyMRkwFwYDVQQDDBBXZWJBRE0g

                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
        <saml:Subject>
            <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:emailAddress">yoan@rcdevs.com</saml:NameID>
            <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
<saml:SubjectConfirmationData InResponseTo="" NotOnOrAfter="2020-10-26T09:27:46Z"
                Recipient="https://signin.aws.amazon.com/saml"/></saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions NotBefore="2020-10-26T09:25:46Z" NotOnOrAfter="2020-10-26T09:27:46Z">
            <saml:AudienceRestriction>
                <saml:Audience>https://signin.aws.amazon.com</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AuthnStatement AuthnInstant="2020-10-26T09:26:46Z" SessionIndex="1">
            <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassR
```

```
        </saml:AuthnContext>
      </saml:AuthnStatement>
      <saml:AttributeStatement>
        <saml:Attribute Name="uid">
          <saml:AttributeValue>administrator</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="domain">
          <saml:AttributeValue>yorcdevs.eu</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="group">
          <saml:AttributeValue>organization management</saml:AttributeValue>
          <saml:AttributeValue>group policy creator owners</saml:AttributeValue>
          <saml:AttributeValue>domain admins</saml:AttributeValue>
          <saml:AttributeValue>enterprise admins</saml:AttributeValue>
          <saml:AttributeValue>schema admins</saml:AttributeValue>
          <saml:AttributeValue>administrators</saml:AttributeValue>
          <saml:AttributeValue>denied rodc password replication group</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">

<saml:AttributeValue>arn:aws:iam::909745736108:role/112345678,arn:aws:iam::909745736108:saml-provider/webadm1.yorcdevs.eu</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/RoleSessionName">
          <saml:AttributeValue>administrator</saml:AttributeValue>
        </saml:Attribute>
        <saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/SessionDuration">
          <saml:AttributeValue>420</saml:AttributeValue>
        </saml:Attribute>
      </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

5.2 Login request on the IDP

The first step is the OpenID login request performed on the OpenID & SAML web application :

5.2.1 OpenID

It starts with :

```
[Mon Oct 26 10:35:53.328922 2020] [192.170.3.23] [OpenID:GTZ09PU0] New login request (OpenOTP)
[Mon Oct 26 10:35:53.328996 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Client ID: OpenID
[Mon Oct 26 10:35:53.329012 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Username: administrator
[Mon Oct 26 10:35:53.329023 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Domain: support
[Mon Oct 26 10:35:53.329035 2020] [192.170.3.23] [OpenID:GTZ09PU0] > ANY Password: xxxxxxxx
[Mon Oct 26 10:35:53.329058 2020] [192.170.3.23] [OpenID:GTZ09PU0] Sending openotpSimpleLogin request
```

The last line of log indicate the login request is sent to OpenOTP. When OpenID call OpenOTP, the session number is the same for the OpenID request and the OpenOTP request (here GTZ09PU0). That allow you to easily identify different requests and products if you need to troubleshoot.

Then, the next part is the OpenOTP request and OpenID request continu after the OpenOTP request.

```
OpenOTP logs available in the next section
```

Below the OpenID session logs after the success login with OpenOTP :

```
[Mon Oct 26 10:35:59.608951 2020] [192.170.3.23] [OpenID:GTZ09PU0] OpenOTP authentication success
[Mon Oct 26 10:35:59.609206 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved LDAP user: CN=Administrator,CN=Users,DC=yorcdevs,DC=eu (cached)
[Mon Oct 26 10:35:59.609399 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved LDAP groups: organization management,group policy creator owners,domain admins,enterprise admins,schema admins,administrators,denied rodc password replication group
[Mon Oct 26 10:35:59.609660 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved source location: US
[Mon Oct 26 10:35:59.622375 2020] [192.170.3.23] [OpenID:GTZ09PU0] Login session started for CN=Administrator,CN=Users,DC=yorcdevs,DC=eu
[Mon Oct 26 10:35:59.830787 2020] [192.170.3.23] [OpenID:GTZ09PU0] Enforcing client policy: Amazon Web Service
[Mon Oct 26 10:35:59.830849 2020] [192.170.3.23] [OpenID:GTZ09PU0] Returning nameId value: 'support@rcdevs.com'
[Mon Oct 26 10:35:59.847865 2020] [192.170.3.23] [OpenID:GTZ09PU0] Sent SAML login success response
```

That part of the logs are important. It shows you the matching with the client policy previously created and the NameID value retuned.

5.2.2 OpenOTP

```
[Mon Oct 26 10:35:53.337483 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] New openotpSimpleLogin SOAP request
[Mon Oct 26 10:35:53.337509 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Username: administrator
[Mon Oct 26 10:35:53.337516 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Domain: support
[Mon Oct 26 10:35:53.337525 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Password: xxxxxxxx
```

```
[Mon Oct 26 10:35:53.337531 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Client ID: OpenID
[Mon Oct 26 10:35:53.337537 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Source IP: 192.170.3.23
[Mon Oct 26 10:35:53.337543 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Context ID: 578d78fb7b15a258ea414ffa9db4ebb2
[Mon Oct 26 10:35:53.337601 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Registered openotpSimpleLogin request
[Mon Oct 26 10:35:53.338238 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved LDAP user: CN=Administrator,CN=Users,DC=yorcdevs,DC=eu (cached)
[Mon Oct 26 10:35:53.338472 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved LDAP groups: organization management,group policy creator owners,domain admins,enterprise admins,schema admins,administrators,denied rodc password replication group
[Mon Oct 26 10:35:53.338718 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved source location: US
[Mon Oct 26 10:35:53.358316 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Started transaction lock for user
[Mon Oct 26 10:35:53.370983 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found user fullname: Administrator
[Mon Oct 26 10:35:53.371005 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found user language: EN
[Mon Oct 26 10:35:53.371018 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 user mobiles: 123456
[Mon Oct 26 10:35:53.371025 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 user emails: support@rcdevs.com
[Mon Oct 26 10:35:53.371467 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 48 user settings:
LoginMode=LDAPOTP,OTPType=TOKEN,OTPFallback=MAIL,PushLogin=Yes,ChallengeMode=Yes,ChallengeTi
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=U2F,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,La
[5 Items]
[Mon Oct 26 10:35:53.372017 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 5 user data: TokenType,TokenKey,TokenState,TokenID,TokenSerial
[Mon Oct 26 10:35:53.372085 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 registered OTP token (TOTP)
[Mon Oct 26 10:35:53.372112 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Requested login factors: LDAP & OTP
[Mon Oct 26 10:35:53.382710 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] LDAP password Ok
[Mon Oct 26 10:35:53.383006 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Authentication challenge required
[Mon Oct 26 10:35:53.564385 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Sent push notification for token #1
[Mon Oct 26 10:35:53.564427 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Waiting 28 seconds for mobile response
[Mon Oct 26 10:35:59.598111 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] Received mobile authentication response from 192.170.3.27
[Mon Oct 26 10:35:59.598145 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] > Session: QIO1HmdExVHo9kr1
[Mon Oct 26 10:35:59.598152 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] > Password: 16 Bytes
[Mon Oct 26 10:35:59.598158 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] Found authentication session started 2020-10-26 10:35:53
[Mon Oct 26 10:35:59.598252 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] PUSH password Ok (token #1)
[Mon Oct 26 10:35:59.605533 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Updated user data
[Mon Oct 26 10:35:59.607544 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Sent login success response
```