HCL MT CH-MSP Product Documentation

Keycloak Authentication and SSO

SSO and Authenticate using Customer IdP

- **Config keycloak to use Customer IdP using SAML**

Configure KeyCloak as OIDC provider for Connections

SSO between Connections, SameTime and Domino

Configure KeyCloak to connect to an LDAP

Configure WAS as OIDC RP

- **Configure WAS as OIDC RP in multi-clusters env**
- **multi-clusters- make sure set- Trusted authentication realms - outbound**

Configure Connections Application user roles

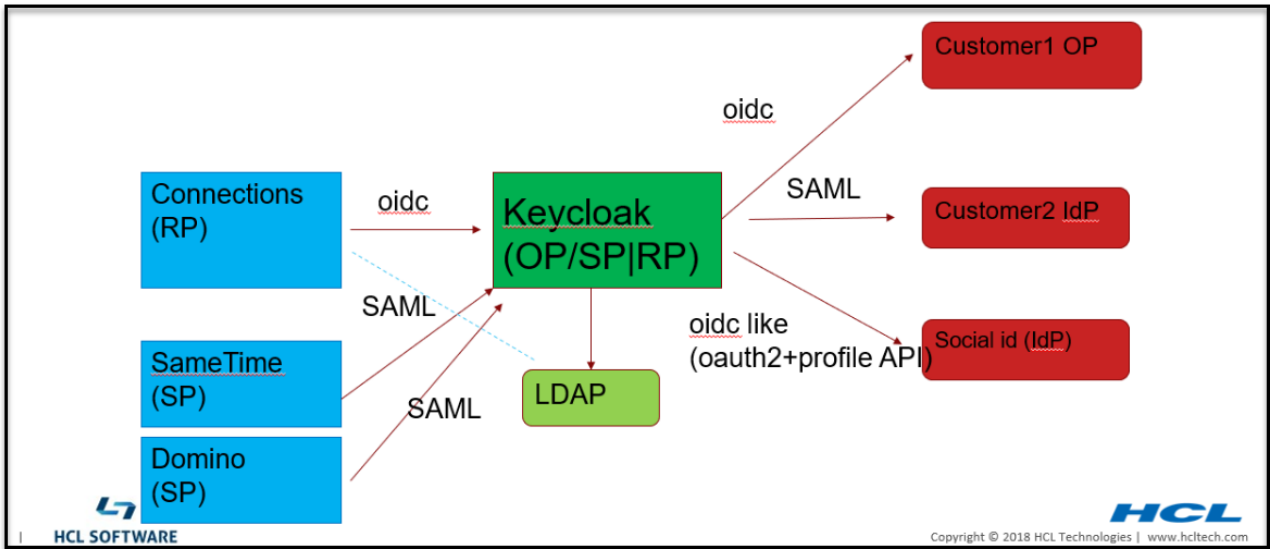
update reverse proxy to handle some redirects

Support OAuth 2 for internal apps and 3rd party apps

- **Allowing third-party applications access to data via the OAuth2 protocol**
- **Enable EE using OIDC OP as OAuth2 provider**
- **Enable RTE using OIDC OP as OAuth Provider**

Keycloak Authentication and SSO

Note: the dot line from connections to ldap is not required for authentication. It may be used for group support from application level. Although it should really be done using keycloak APIs, the existing code may be doing a direct ldap call.



In MT, most of customers have their own IdP to authenticate users so Connections can SSO to customers own applications that use the same IdP.

We can use Keycloak to connect to customer Idp either via SAML or OIDC.

Identity Providers

Name	Provider	Enabled	Hidden	Link only	GUI order
oidc	oidc	True	False	False	
orgf	saml	True	False	False	
google	google	True	False	False	

By default, user would have to pick which login to use. However we can remove this page by redirect the user to their org's IdP.

This can be done by using **`kc_idp_hint`**. see
https://www.keycloak.org/docs/latest/server_admin/#_client_suggested_idp

Because redirect URL contains org url, so we can create rewrite rule to map between org to their own IdP.

https://\

{KK}/realms/{RM}/protocol/openid-connect/auth?xxxxx&redirect_uri=https%3A%2F%2Fmtdemo1-orgc.cnx.cwp.pnp-hcl.com%3A443%2Foidcclient_apps%2Fkeycloak&yyyyy

to

https://\{KK}

/realms/{RM}/protocol/openid-connect/auth?xxxxx&redirect_uri=https%3A%2F%2Fmtdemo1-orgc.cnx.cwp.pnp-hcl.com%3A443%2Foidcclient_apps%2Fkeycloak&yyyyy&kc_idp_hint=mtdemo1-orgc

Here is an example implementation:

```
if ($arg_redirect_uri ~ ^(https.*connmt-orge.*)){
rewrite ^(/auth/.*)/Azure-OIDC/(\w+\.?.*$) $1/Azure-OIDC/$2?kc_idp_hint=google break;
}

if ($arg_redirect_uri ~ ^(https.*connmt-orgf.*)){
rewrite ^(/auth/.*)/Azure-OIDC/(\w+\.?.*$) $1/Azure-OIDC/$2?kc_idp_hint=connmt-orgf break;
    }
```

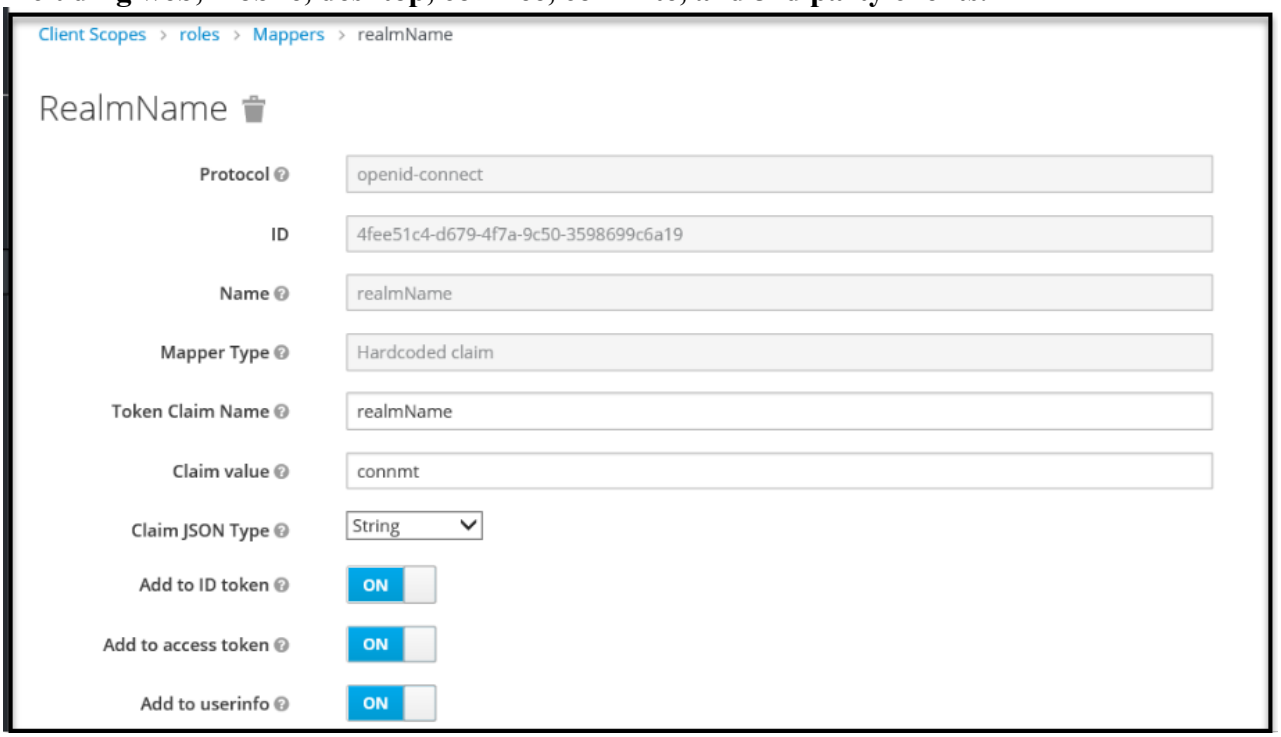
Config keycloak to use Customer IdP using SAML

Azure AD as SAML IdP

Configure KeyCloak as OIDC provider for Connections

Create client for Connections web client

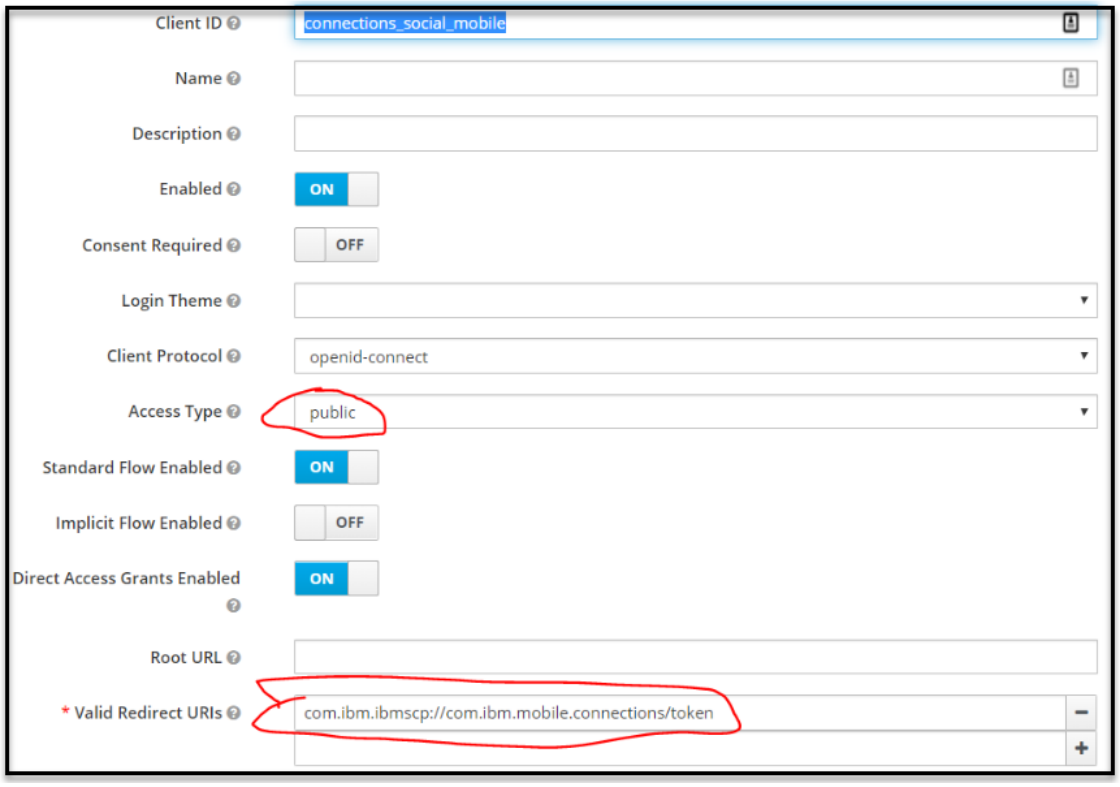
-
- Create a realm
- Connect to the LDAP that contains all Connections users
- Create an OIDC client, type: **confidential**
- In Mapper add properties name: realmName, hardcode the value as your realm name
    - **Note, you can add the mapper in the client scope so it will be available to all clients in the same realm including web, mobile, desktop, conn-ee, conn-rte, and 3rd party clients.**



- call back urls:
    - https://<connections host>/oidcclient/<provider_1.identifier value>
    - Note: for MT also add each org's url, e.g.
        - https://<connections host>_orga/oidcclient/<provider_1.identifier value>
        - https://<connections host>_orgb/oidcclient/<provider_1.identifier value>

Create client for Connections Mobile client

In the same realm, create another client for Connections Mobile:

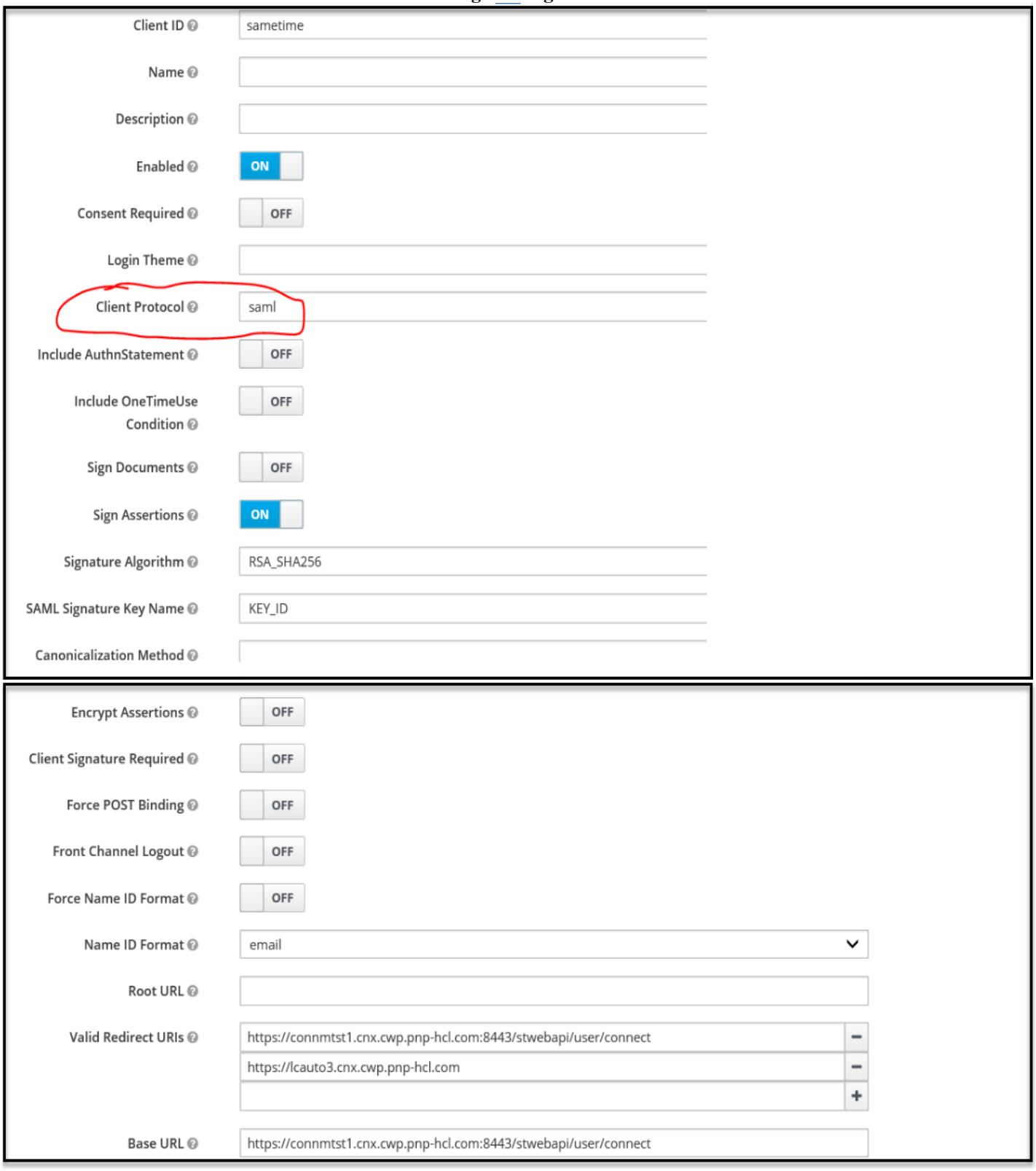


SSO between Connections, SameTime and Domino can be done via Keycloak.

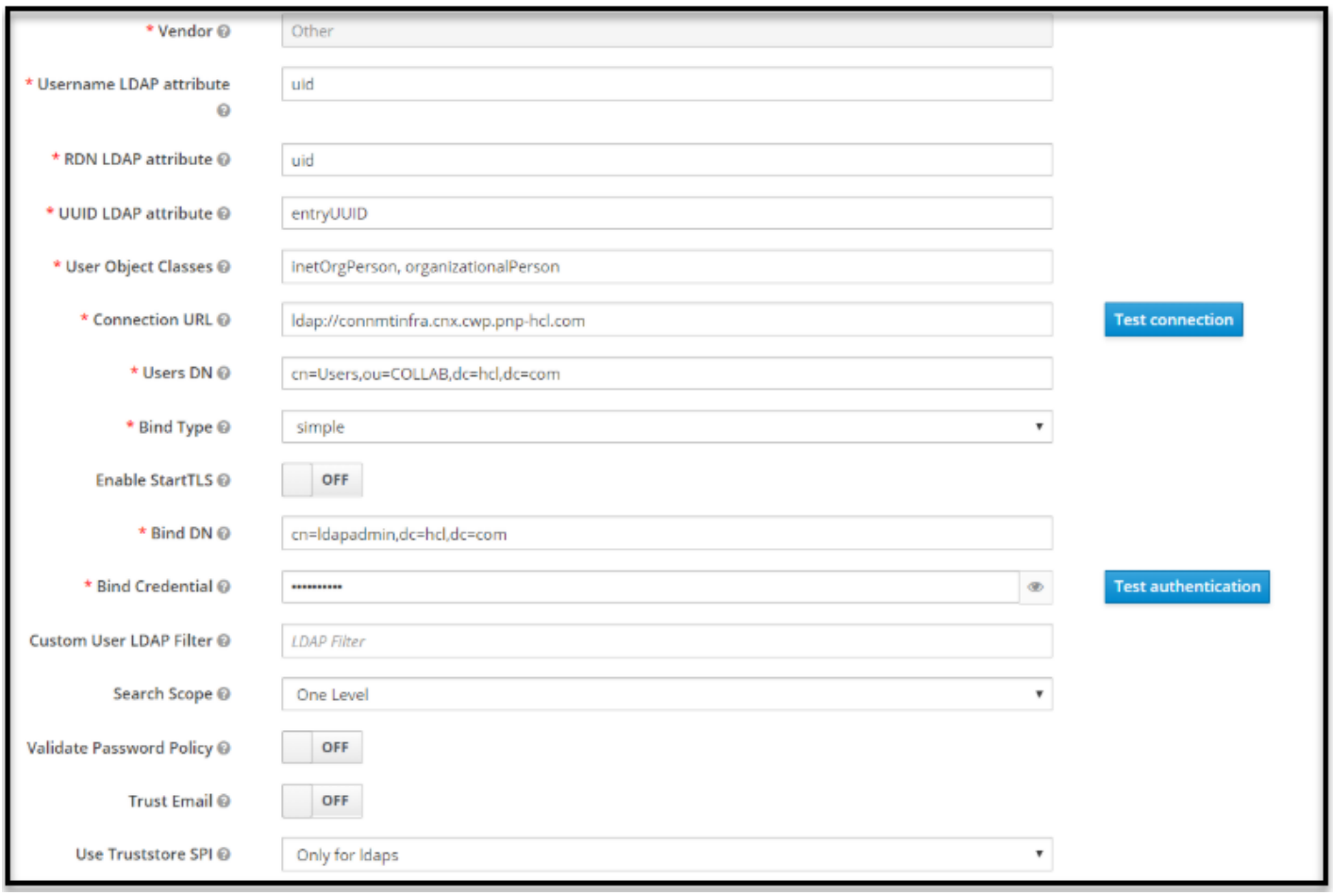
All apps should be in the same realm.

Create each client for each app.

The client for SameTime: use IdP initiated Post binding with signed assertion:



Configure KeyCloak to connect to an LDAP



Configure WAS as OIDC RP

- Install OIDC RP via WAS admin command follow websphere
- documentation configure OIDC TAI properties:

excludedpathFilter:
~~/ibm/console,/ibm/console/.*,/profiles/dsx/.*,/communities/dsx/.*,/dm,/dm/atom/seedlist,/dm/atom/communities/feed,/activities/service/atom2/forms/communityEvent,/communities/recomm/handleEvent,/communities/calendar/handleEvent,/profiles/wdp/*~~

/ibm/console,/ibm/console/.*,/profiles/dsx/.*,/communities/dsx/.*,/dm,/dm/atom/seedlist,/dm/atom/communities/feed,/activities/service/atom2/forms/communityEvent,/communities/recomm/handleEvent,/communities/calendar/handleEvent,/profiles/s

eedlist/myserver note, the services path may not be the same per deployment.

To support JWT as access token for oauth add the following:

provider_1.verifyIssuerInIat=true

provider_1.audiences="connections_social_mobile","account","connmt

" connmt is the client id for Connections web.

onnections_social_mobile is the client id for Connections Mobile

To support Mobile/oauth2 client also be able to use session cookie, added:

provider_1.setLtpaCookie=true

☐	provider_1.identifier	keycloak
☐	provider_1.clientId	connmt
☐	provider_1.clientSecret	a170bb09-9c16-4198-b0a0-3e1bec581ced
☐	provider_1.signatureAlgorithm	RS256
☐	provider_1.scope	openid
☐	provider_1.interceptedPathFilter	/.*
☐	provider_1.excludedPathFilter	/survey/.*,/surveys/.*,/ibm/console,/ibm/console/.*,/profiles/dsx/.*,/communities/dsx/.*,/dm,/dm/atom/seedlist,/dm/atom/communities/feed,/activities/service/a
☐	provider_1.authorizeEndpointUrl	https://lcauto3.cnx.cwp.pnp-hcl.com/auth/realms/connmt/protocol/openid-connect/auth
☐	provider_1.tokenEndpointUrl	https://lcauto3.cnx.cwp.pnp-hcl.com//auth/realms/connmt/protocol/openid-connect/token
☐	provider_1.introspectEndpointUrl	https://lcauto3.cnx.cwp.pnp-hcl.com/auth/realms/connmt/protocol/openid-connect/token/introspect
☐	provider_1.signVerifyAlias	lcauto3
☐	provider_1.jwkEndpointUrl	https://lcauto3.cnx.cwp.pnp-hcl.com/auth/realms/connmt/protocol/openid-connect/certs
☐	provider_1.defaultRealmName	connmt
☐	provider_1.issuerIdentifier	https://lcauto3.cnx.cwp.pnp-hcl.com/auth/realms/connmt
☐	provider_1.userIdentifier	email
☐	provider_1.useJwtFromRequest	ifPresent
☐	provider_1.createSession	true
☐	provider_1.verifyIssuerInIat	true
☐	provider_1.audiences	ALL_AUDIENCES
☐	provider_1.setLtpaCookie	true

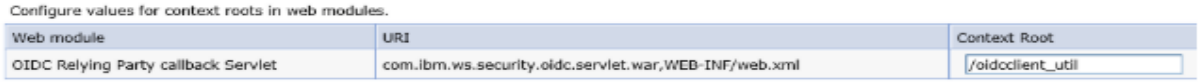
Configure WAS as OIDC RP in multi-clusters env

Connections medium and large deployment consists multiple clusters(JVMs) and each contain number of applications. Due to limitation of WebSphere OIDC RP, the RP stores state in local JVM, hence the callback has to return to the same JVMs where application login started. We have request IBM to fix this via this request: https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=104320 Please help by voting it.

Here is the workaround:

1. Deploy OIDC_RP ear to **each JVM/cluster** with unique context root.

☐	WebSphereOIDCRP	Base edition	Active	Java 2 Platform, Enterprise Edition	➔
☐	WebSphereOIDCRP_apps.ear	Base edition	Active	Java 2 Platform, Enterprise Edition	➔
☐	WebSphereOIDCRP_infra.ear	Base edition	Active	Java 2 Platform, Enterprise Edition	➔
☐	WebSphereOIDCRP_push.ear	Base edition	Active	Java 2 Platform, Enterprise Edition	➔



Configure values for context roots in web modules.

Web module	URI	Context Root
OIDC Relying Party callback Servlet	com.ibm.ws.security.oidc.servlet.war,WEB-INF/web.xml	/oidcclient_util

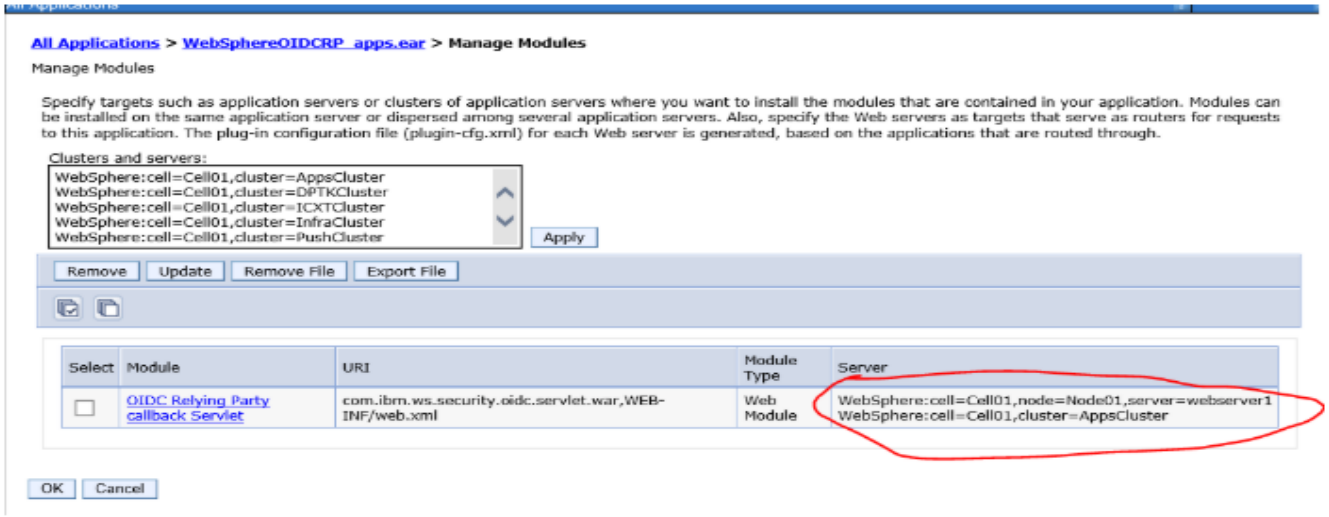
All Applications > WebSphereOIDCRP_apps.ear > Context Root For Web Modules

Context Root For Web Modules

Configure values for context roots in web modules.

Web module	URI	Context Root
OIDC Relying Party callback Servlet	com.ibm.ws.security.oidc.servlet.war,WEB-INF/web.xml	/oidcclient_apps

OK Cancel



All Applications > WebSphereOIDCRP_apps.ear > Manage Modules

Manage Modules

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=Cell01,cluster=AppsCluster
WebSphere:cell=Cell01,cluster=DPTKCluster
WebSphere:cell=Cell01,cluster=ICXTCluster
WebSphere:cell=Cell01,cluster=InfraCluster
WebSphere:cell=Cell01,cluster=PushCluster

Apply

Remove Update Remove File Export File

Select	Module	URI	Module Type	Server
☐	OIDC Relying Party callback Servlet	com.ibm.ws.security.oidc.servlet.war,WEB-INF/web.xml	Web Module	WebSphere:cell=Cell01,node=Node01,server=webserver1 WebSphere:cell=Cell01,cluster=AppsCluster

OK Cancel

2. Configure OIDC RP TAI with a provider for each cluster and intercept the apps with the corespondent provider. **(note: all properties values are the same for each provider except the interceptedPathFilter and callbackServletContextPath)**

intercept path: (note in your environment the app may be deployed on different cluster and please adjust accordingly)

provider_1..interceptedPathFilter: /push/.*

provider_2.interceptedPathFilter:
/connections/bookmarklet/.*,/connections/oauth/.*,/connections/resources/.*,/connections/config/.*,/communities/.*,/connections/proxy/.*,/help/.*,/xcc/.*,/selfservice/.*,/news/.*,/profiles/.*,/search/.*,/socialsidebar/.*,/touchpoint/.*,/connections/thumbnail/.*,/connections/opengraph/.*,/oauth2/.*,/connections/opensocial/.*

provider_3..interceptedPathFilter: /homepage/.*,/moderation/.*,/connections/rte/.*,/connections/webeditors/.*

provider_4..interceptedPathFilter: /activities/.*,/blogs/.*,/dogear/.*,/files/.*,/forums/.*,/metrics/.*,/metricssc/*,/mobile/.*,/connections/filesync/.*,/connections/filediff/.*,/mobileAdmin/.*,/storageproxy/.*,/wikis/.*

Select	Name	Value
☐	provider_1.identifier	keycloak
☐	provider_1.clientId	cdemo1
☐	provider_1.clientSecret	231e9d75-622e-4d04-88d7-cbfdeabca3e5
☐	provider_1.signatureAlgorithm	RS256
☐	provider_1.scope	openid
☐	provider_1.interceptedPathFilter	/activities/.*,/blogs/.*,/dogear/.*,/files/.*,/forums/.*,/metrics/.*,/metricssc/.*,/storageproxy/.*,/wikis/.*,/mobile/.*,/connections/filesync/.*,/connections/filediff/.*
☐	provider_1.excludedPathFilter	/ibm/console,/ibm/console/.*,/profiles/dsx/.*,/communities/dsx/.*,/dm,/dm/atom/seedlist,/dm/atom/communities/feed,/activities/service/atom2/forms/community
☐	provider_1.authorizeEndpointUrl	https://mtdemo1.cnx.cwp.pnp-hcl.com/auth/realms/connectdemo/protocol/openid-connect/auth
☐	provider_1.tokenEndpointUrl	https://mtdemo1.cnx.cwp.pnp-hcl.com/auth/realms/connectdemo/protocol/openid-connect/token
☐	provider_1.introspectEndpointUrl	https://mtdemo1.cnx.cwp.pnp-hcl.com/auth/realms/connectdemo/protocol/openid-connect/token/introspect
☐	provider_1.signVerifyAlias	mtdemo1
☐	provider_1.jwkEndpointUrl	https://mtdemo1.cnx.cwp.pnp-hcl.com/auth/realms/connectdemo/protocol/openid-connect/certs
☐	provider_1.defaultRealmName	connectdemo
☐	provider_1.issuerIdentifier	https://mtdemo1.cnx.cwp.pnp-hcl.com/auth/realms/connectdemo
☐	provider_1.userIdentifier	email
☐	provider_1.createSession	true
☐	provider_1.verifyIssuerInIat	true
☐	provider_1.audiences	ALL_AUDIENCES
☐	provider_1.useJwtFromRequest	ifPresent
☐	provider_1.setLtpaCookie	true
☐	provider_1.callbackServletContext	/oidcclient_apps

4. enable custom dynacache.

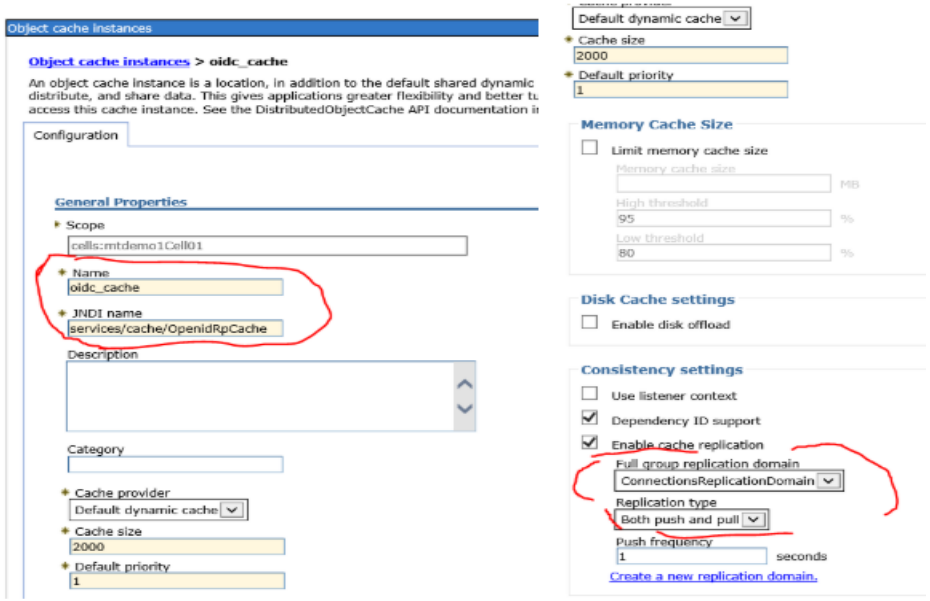
4,1 In oidc RP TAI properties add: jndiChaneName:

☐	provider_4.interceptedPathFilter	/activities/.*,/blogs/.*,/dogear/.*,/files/.*,/forums/.*,/m
☐	jndiCacheName	services/cache/OpenidRpCache

4.2: Create a new object cache instance with the JNDI name match the one use in the TAI property

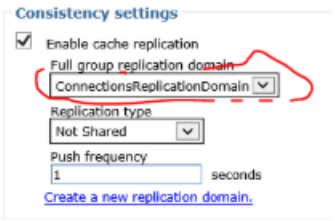
above. replication Domain: **ConnectionsReplicationDomain**

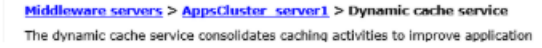
replication Type: **both push and pull**



4.3

In **each cluster (Apps, Infra, util, push)** Dynamic chache service make sure cache replication is enabled and is using ConnectionsReplicaitonDomain.



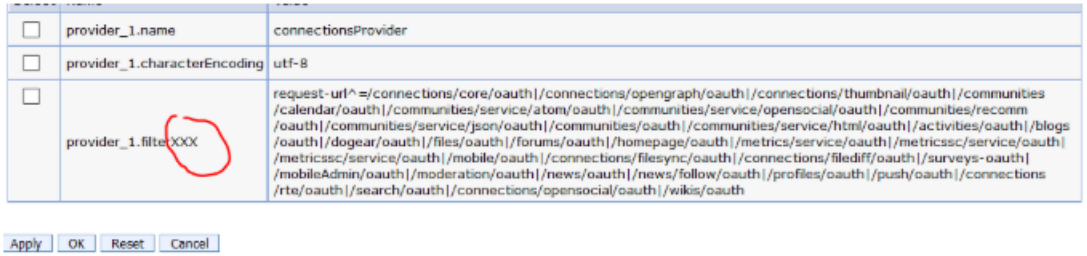


5. update callbacks in keycloak with content root.

6. Custom properties, make sure removeboth

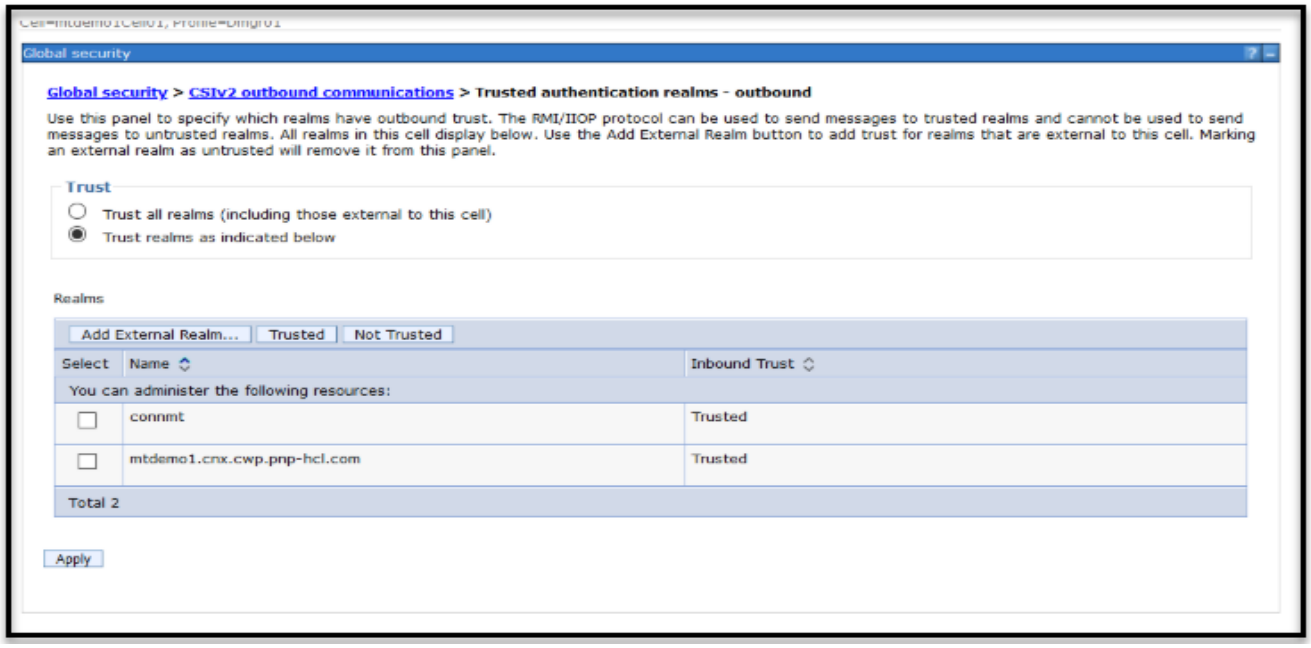
com.ibm.websphere.security.DeferTAItoSSO
com.ibm.websphere.security.InvokeTAIbeforeSSO

7. set oauth2 tai filter to some dummy value so it won't intercept any request. e.g.



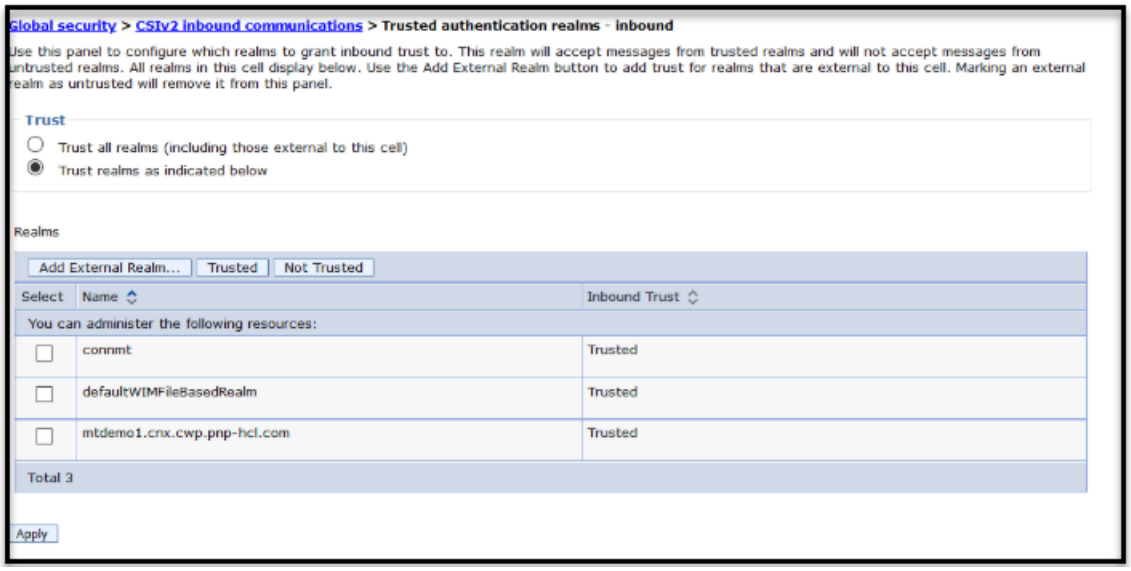
multi-clusters- make sure set- Trusted authentication realm

make sure set Global security > CSIv2 outbound communications > Trusted authentication realms - outbound to "Trust realms as indicated below" and include the realm you defined in keycloak mapper, see screenshot below.



Note:

Global security > CSIv2 inbound communications >
Trusted authentication realms – **inbound** has already set and should be the same as when you set it from globe Security.



Configuring Users and Administrators in Connections:

1. all authenticated users

In all Connections applications change all the rows that read "All Authenticated in Application's Realm" to read "All Authenticated in Trusted Realms" (found in the Map Special Subjects dropdown).

2. admin or special users/group access

For example typical the rows that contain "ajones1" on our Connections test machines. Select the row and click: "Map Users...". Then change "User Realm" to "<keyckoak's realmname>" and search for ajones1. In the form that opens enter "ajones1" for User short name and email for Unique user ID. Transfer the user to the selected list and click "OK". Then save your changes to the Websphere configuration. Sync the notes and restart the server.

For example:

These users are mapped as ICEC Admins ajones250@janet.iris.com@connmt
suser1@janet.iris.com@connmt

where ajones250@janet.iris.com@connmt is orgb admin, and suser1@janet.iris.com@connmt is orga admin

Update reverse proxy to handle some redirects

Adding Rewrite Rules in Reverse Proxy:

Some Connections login urls are not protected, they will not be intercepted by OIDC Provider, we need to add Rewirte Rule in reverse proxy to make the browser redirect to protected url.

1. go to /opt/IBM/HTTPServer/conf
2. edit file ihs-upload-rewrite.conf
3. add following rules:

 # mt.install.cfg.start
Redirect /communities/login /communities/service/html/login
Redirect /homepage/login /homepage/
Redirect /homepage/auth/login.jsp /homepage/
Redirect /activities/auth/login.jsp /activities
Redirect /profiles/login /profiles/html/myProfileView.do
RedirectMatch /profiles/profile.do(.) /profiles/html/myprofile.do$1*
Redirect /forums/auth/login /forums/html/my
Redirect /blogs/login /blogs/roller-ui/myblogs/edit
Redirect /mobileAdmin/login /mobileAdmin/console

 # OIDC discovery for the backend Keycloak OIDC server
Redirect "/.well-known/openid-configuration" "https://lcauto3.cnx.cwp.pnp-hcl.com/auth/realms/connmt/.well-known/openid-configuration"
 # mt.install.cfg.end

1. go to /opt/IBM/HTTPServer/bin

   run command sudo apachect1 restart

Support OAuth 2 for internal apps and 3rd party apps

In the MT environment Keycloak/OIDC will be the OAuth2 provider for both internal apps, external apps access to Connections data and Connections Mobile, Desktop plugins etc.

Note:

Because some internal apps such as RTE and Embedded Experience (EE) use Oauth2 to access to Connections data, to avoid to have the user login again, the oauth2 dance needs to carry the authentication cookies back to Keycloak during authorization process, the keycloak authentication cookies must have the same domain and path as Connections domain and context path so these cookies are visible to Connections applications.

LtpaToken2	I1I8S0ROs5QwM/T5N+NWnik8nOZHo8uBM0LzMpAs0afWp3FVO0f12MmMZy2...	.cnx.cwp.pnp-hcl	758 B	/
OIDCSESSIONID_lcauto130	551295430	lcauto130.cnx.c...	32 B	/
KEYCLOAK_SESSION	poolrealm/6d265c22-2734-4e8d-8707-27edbeffd95c/7497792b-3933-4b33-9b9c-	.cnx.cwp.pnp-hcl	99 B	/
KEYCLOAK_IDENTITY	eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI1ZDBmZGQ0YS0zZGI3L...	.cnx.cwp.pnp-hcl	634 B	/
AUTH_SESSION_ID	7497792b-3933-4b33-9b9c-0fbcf5ff677f.lcauto3	.cnx.cwp.pnp-hcl	56 B	/
OIDCSTATE_lcauto130	rO0ABXNyABNqYXZhLnV0aWwuSGFzaHRhYmxlE7sPJSFK5LgDAAJGAApsb2...	lcauto130.cnx.c...	484 B	/
ICSESSIONID	0000ocUoOWNMABc_if3vRtkc-Gu:-1	lcauto130.cnx.c...	41 B	/
lcLang	en_us	lcauto130.cnx.c...	11 B	/

This can be done by a proxy rule.

Allowing third-party applications access to data via the OAuth2 protocol

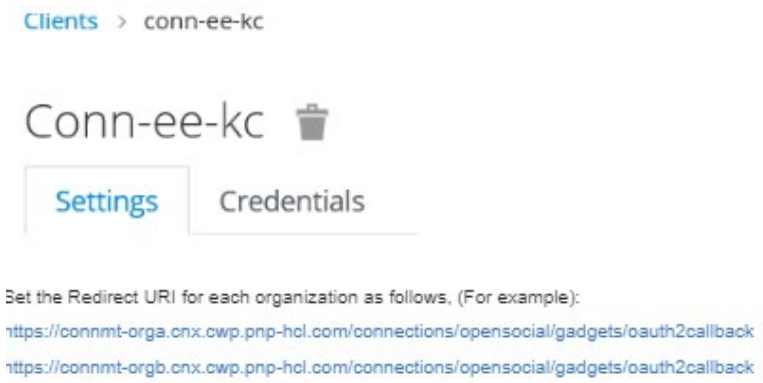
1. MSP provide initial access tokens to their customer administrators.

2. Customer can use this access token to create client for their applications to access to Connections data.

Enable EE using OIDC OP as OAuth2 provider

This article describes how to properly setup the Connections Embedded Experience client at Keycloak.

1. Register a client on Keycloak, my example is conn-ee-kc.

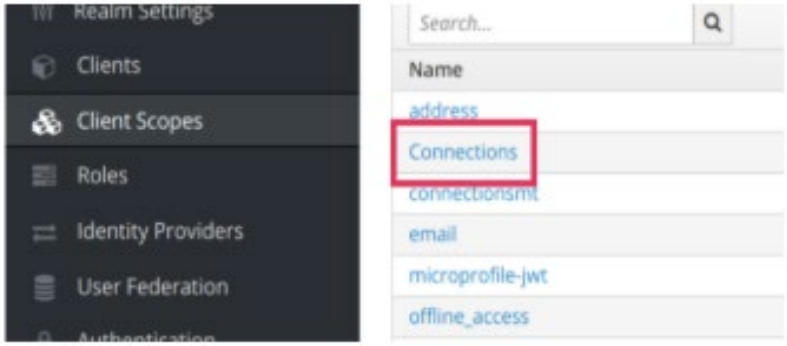


Set the Redirect URI for each organization as follows, (For example):

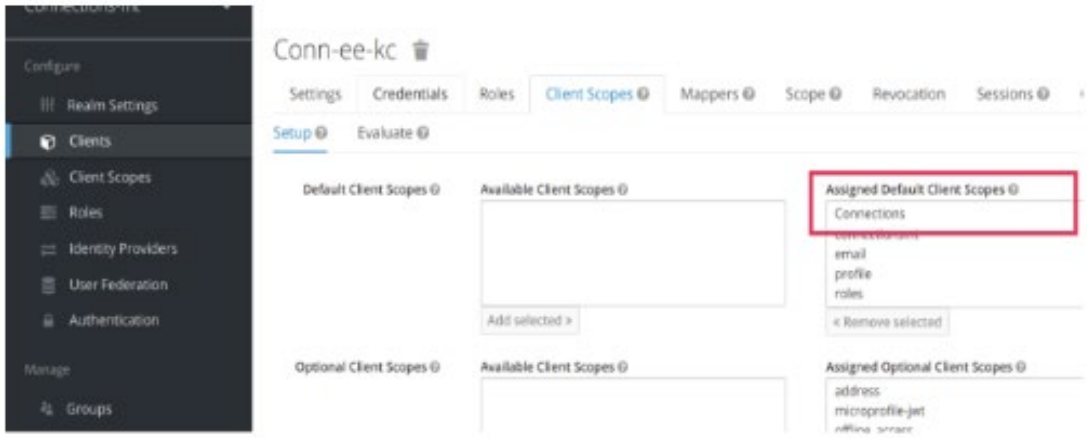
https://connmt-orga.cnx.cwp.pnp-hcl.com/connections/opensocial/gadgets/oauth2callback

https://connmt-orgb.cnx.cwp.pnp-hcl.com/connections/opensocial/gadgets/oauth2callback

2. Add a Keycloak Client Scope called Connections for your realm.



3. Associate the Connections scope as a default client scope for the Conn-ee-kc client



4. (This step Should not be required if running the mtupdate scripts!) run the (note, replace client secret, keycloakauth and token endpoints with yours.

register_client_oidc_connmt.py

5. (This step Should not be required if running the mtupdate scripts!) create a proxy-policy.dynamic under LotusConnections-config/opensocial-proxy-rules directory

add

allow('.', '.', 'https:\/\/lcauto3.cnx.cwp.pnp-hcl.com\/auth\/realms\/poolrealm\/protocol\/openid-connect\/token.*');

Note,

- change the realm name based on the env. for example, for "connmt" realm, change "poolrealm" to "connmt",

- change host to your keycloak host.

Enable RTE using OIDC OP as OAuth Provider

1. replace RichTextEditors.ear

2. register **conn-rte** client in keycloak. with callback <connections_host>/connections/rte/connect For example, the callback URI for lcauto130 is https://lcauto130.cnx.cwp.pnp-hcl.com/connections/rte/connect.

3.Create **oidcRTEClientAuth** J2C alias in WebSphere (Global Security -> Java Authentication andAuthorization Services -> J2C authentication data)

Add an alias with the following values:
Alias is **oidcRTEClientAuth**
User ID is **conn-rte**
Password is the client_secret

4. Modify service-location.xsd and add the following to the list of serviceNames:

<xsd:enumeration value="oidc_op" />

5. Modify LotusConnections-config.xml and add the following serviceReference, replacing YOUR_REALM_NAME and YOUR_KEYCLOAK_SERVER appropriately:

```
   <sloc:serviceReference bootstrapHost="admin_replace" bootstrapPort="admin_replace" clusterName=""
enabled="true" serviceName="oidc_op" ssl_enabled="true">
      <sloc:href>
         <sloc:hrefPathPrefix>/auth/realms/YOUR_REALM_NAME/.well-known/openid-
configuration</sloc:hrefPathPrefix>
         <sloc:static href="http://YOUR_KEYCLOAK_SERVER.cnx.cwp.pnp-hcl.com"
ssl_href="https://YOUR_KEYCLOAK_SERVER.cnx.cwp.pnp-hcl.com"/>
         <sloc:interService href="https://YOUR_KEYCLOAK_SERVER.cnx.cwp.pnp-hcl.com"/>
      </sloc:href>
   </sloc:serviceReference>
```

For example, on lcauto130 I'm authenticating against the poolrealm on lcauto3's keycloak server:

```
   <sloc:serviceReference bootstrapHost="admin_replace" bootstrapPort="admin_replace" clusterName=""
enabled="true" serviceName="oidc_op" ssl_enabled="true">
      <sloc:href>
         <sloc:hrefPathPrefix>/auth/realms/poolrealm/.well-known/openid-configuration</sloc:hrefPathPrefix>
         <sloc:static href="http://lcauto3.cnx.cwp.pnp-hcl.com" ssl_href="https://lcauto3.cnx.cwp.pnp-hcl.com"/>
         <sloc:interService href="https://lcauto3.cnx.cwp.pnp-hcl.com"/>
      </sloc:href>
   </sloc:serviceReference>
```

Note, step 4 and 5 need to be done when server is stopped.

restart server after all the changes.