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.

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, keycloak auth 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 and Authorization 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.