HCL MT CH-MSP Product Documentation

Configuring the Multi-Tenant environment

Configuring the Multi-Tenant environment

Topics include:

- Import the Keycloak SSL certificate Running the auto deploy scripts
- Validating the scripts
- Post installation steps
- Configure OrientMe web client container for Multi-Tenant

Import the Keycloak SSL certificate

The WebSphere Open ID Connect Relying Party TAI requires that the public signing certificate from the OIDC provider is imported into the Cell Default Trust store in WAS. The OIDC RP uses this certificate to validate that a JSON Web Token provided by a client is authentic. The alias of this certificate is used during the OIDC TAI definition, using key **verifySignAlias**. You should import this certificate into WAS prior to running the MT scripts.

To import the keycloak certificate, follow these steps using the WebSphere Administration console:

1. Navigate to Security -> SSL Certificate and key Management -> Key stores and certificates -> CellDefaultTrustStore
2. Select Signer certificates
3. Select Retrieve from port
4. Enter the keycloak hostname and port
5. Take note of the alias, such as **oidc-provider**, as this is used later.
6. Select Retrieve signer information and review the certificate information.
7. Select OK and then synchronize nodes.

Use the value of the alias as the value for **mt.oidc.signVerifyAlias** which is defined in the config.properties file used by the mtupdate script.

How to use

Dependencies

1. This installer requires an existing and functioning HCL Connections 6.5 CR1a installation.
2. The LDAP/user directory connected to the Connections installation should be prepared according to MT requirements (e.g. organization trees containing users)
3. The Keycloak OIDC provider should be installed and configured for the environment. The necessary configuration properties (clientId, clientSecret, etc.) should be noted as they are required as input for the installer.

Preparing the installer

1. Copy the installation zip to the Deployment Manager server of your Connections environment at your preferred location

   connections-mt-update-6.5.0.zip|View Details

2. Extract the archive

```
> unzip connections-mt-update-6.5.0.zip
> ls
connections-mt-update-6.5.0.zip deployment-units install META-INF  version.txt
```

- **Note**: Please leverage a user with appropriate access rights - i.e. one that has similar priviledges as the configured WebSphere user

3. Navigate into the /install directory and open the `config.properties` file with a text editor of your choice

```
> cd install/
> ls
autodeploy-install-cli.sh config.properties connections-mt-init-entitlements-cli.sh connections-mt-update-cli.sh connections-
mt-webserver-update-cli.sh  resources
> vim config.properties
```

4. Fill in the environment-specific properties for each of the configuration parameters. Use the comments as guidance, e.g. by comparing the samples with your existing installation. **Make sure you completely review and fill out every key in config.properties. This is a critical step!** See the end of this article for more information on the properties.

5. Make shell scripts executable

```
> chmod +x *.sh
```

Installing AutoDeploy v5.3.0+

The installation routine takes advantage of HCL AutoDeploy (https://help.hcltechsw.com/connections/api/assets/autodeploy-html/index.html#) capabilities, e.g. to update existing configuration files. For this, AutoDeploy in version 5.3.0 or higher has to be installed on the environment. If this is already the case the installation can be skipped. The AutoDeploy related configuration properties have to be properly prepared in either case.

To install AutoDeploy, please leverage the provided script `autodeploy-install-cli.sh`

```
> ./autodeploy-install-cli.sh config.properties 2>&1 | tee autodeploy_install.log
```

- **Note**: If you have a previous version of AutoDeploy installed, please install at a different location or back up your installation, as this will overwrite your existing configuration

AutoDeploy requires OS user credentials or a ssh key file (depending on your environment) in order to log in and execute tasks on behalf of the WebSphere administrative user. Please make sure the tool is appropriately prepared to execute the installation with. If there are any issues, the above installation script should already end in some error indicating what may need to be adjusted. To ensure AutoDeploy will be working, run the command `[PATH_TO_AD_INSTALLATION]/app/console.sh --profilepath [PATH_TO_DMGR_PROFILE]` and check if AutoDeploy starts up without issues. If ssh keys are required, please add the parameters `--pubkey --keyfile=[PATH_TO_KEYFILE]` to the command.

- **Note**: In case of leveraging an SSH key, please make sure the key is added the the `authorized_keys` of the respective OS user. If the installation environment consists of multiple nodes, the key has to be authorized for all nodes.

Running the installer

Run the updater via provided script `connections-mt-update-cli.sh`. It is recommended to pipe all of the output to a log file for review.

```
> ./connections-mt-update-cli.sh config.properties 2>&1 | tee mtupdate.log
```

This script will run through various update tasks and will take the configuration file as guidance on where to find or how to update properties and files. Make sure the values reflect your environment.

There may be some of the following prompts during installation:

- User/password prompt in AutoDeploy - see previous notes to AutoDeploy installation
- Whether the WebSphere Cell should be restarted
  - this is necessary in order to execute tasks that require the Connections instance be started in a MT configuration setting
  - if the script does not perform the restart, then make sure to synchronize all nodes and restart the WebSphere Deployment Manager, all Nodes and Servers.

Note: The installer will adjust and extend various parts of the Connections installation related to WebSphere, config files, DB2, web servers and installed applications. As the Connections instance commonly is spread across various servers, the installer may be unable to apply certain changes. For this, the installer provides additional scripts that can be executed directly on the respective server:

- `connections-mt-webserver-update-cli.sh` - applies changes to the Webserver instance. Has to be executed on all servers hosting the IHS web server.
- `connections-mt-db-update-cli.sh` - applies changes to the DB2 schemas and config. Has to be executed on the server hosting DB2.
- `connections-mt-appserver-update-cli.sh` - applies changes to Connections components. Has to be executed on all servers hosting Connections nodes.

To apply the necessary changes, the suggested approach is to copy the installation directory from the Deployment Manager to the respective server, and executing the corresponding script. If any values in the configuration file may differ (e.g. different Application Server path/name), they have to be adjusted before running the script.

Regenerate the IBM Http Server plugin

After the Connections server has restarted, regenerate and propagate the web server plugin using the WebSphere Administration console. Follow these steps using the WebSphere Administration console:

- Navigate to Servers > Server Types > Web servers
- Select the webserver and click Generate Plug-in
- Select the webserver and click Propagate Plug-in
- Select the webserver and click Stop
- Select the webserver and click Start

Restart the DB2 server

When running the MT upgrade scripts, the property DB2_SELECTIVITY will be globally enabled at the DB2 server. If this property was not previously set, a manual DB2 server restart is required prior to validating the Connections server. Perform the following steps:

- Stop the Connections server applications
- Stop DB2 (force it down if necessary)
- Start DB2

This step must be run prior to provisioning users.

Provision Connections Users

A sample script has been provided to provision an initial set of users that have been populated in LDAP under `provision.mt.saasOrganizationsLdapDN`. Running the following script will add these users to Connections and entitle them for Connections services.

```
> ./connections-mt-init-entitlements-cli.sh config.properties 2>&1 | tee mtupdate_entitlements.log
```

Configuration properties

This section describes the available/necessary configuration properties as well as what they refer to and their default values

WebSphere and environment properties

Name	Description	Default
DMGR_PROFILE	The path to the DMGR profile	`/opt/IBM/WebSphere/AppServer/profiles/Dmgr01`
DMGR_NAME	The name of the DMGR profile	`Dmgr01`
WAS_HOME	The path to the WebSphere AppServer installation root	`/opt/IBM/WebSphere/AppServer`
APPSRV_PROFILE	The Application Server profile path	`/opt/IBM/WebSphere/AppServer/profiles/AppSrv01`
IHS_HOME	The IBM HTTP Server path	`/opt/IBM/HTTPServer`
DMGR_SOAP_PORT	The SOAP port that is used to access the MBeans of the DMGR process. This value can be found in WAS console/ISC: System Administration > Deployment manager > Additional Properties > Ports	`8879`

CELL_NAME	The cell name of the Connections installation. This value can be retrieved by looking into the config directory: `> ls -l $DMGR_PROFILE/config/cells`	`Cell01`
WASADMIN_USER	The username of the wasadmin user	`wasadmin`
WASADMIN_PASSWORD	The password of the wasadmin user	`password`
CONNECTIONS_PATH	The path to the Connections content folder	`/opt/HCL/Connections`
AUTODEPLOY_HOME	The path to the AutoDeploy installation. If AutoDeploy is not installed on this machine, this path will be used as the installation directory	`/opt/HCL/autodeploy`
AUTODEPLOY_AUTH_METHOD	The method of authentication for AutoDeploy - supported methods are [password, pubkey]	`password`
AUTODEPLOY_AUTH	The means of authentication - this can be either a password or the path to a RSA public key file. Sample: AUTODEPLOY_AUTH=password, AUTODEPLOY_AUTH=/home/user/.ssh/id_rsa. Leaving this blank will mean that the according value (based on AUTODEPLOY_AUTH_METHOD) will be prompted for	NO VALUE SET BY DEFAULT
DB2_USER	Specify the username of the DB2 user. This is the OS user that has access rights to the DB2 WIKIS and FILES database to initiate SQL scripts with	`db2inst1`
DB2_PASSWORD	Specify the password of the DB2 user	`password`

HCL Connections configuration properties

The following properties are required to update Connections configuration files (LotusConnections-config.xml, directory.services.xml, mobile-config.xml)

Name	Description	Default
mt.connections.url	The hostname/domain of the Connections MT installation. E.g. `connmt.mycompany.com`	NO VALUE SET BY DEFAULT
mt.connections.url.parent	The parent domain of the Connections MT installation. E.g. if your Connections MT domain is "connmt.mycompany.com", the parent domain should be "mycompany.com"	NO VALUE SET BY DEFAULT
mt.ldap.url	The URL to the LDAP including protocol and port. E.g. `ldap://connmt-ldap.mycompany.com:389`	NO VALUE SET BY DEFAULT
mt.ldap.security.authentication	The authentication type to be used. Available options are: 1) none = no authentication, 2) simple = authenticating via user credentials	`none`
mt.ldap.security.principal	The user principal to be used for LDAP authentication. E.g. cn=root,dc=mycompany,dc=com	NO VALUE SET BY DEFAULT
mt.ldap.baseDN	The base DN entities are added to in the user directory. E.g. ou=collab,dc=mycompany,dc=com	NO VALUE SET BY DEFAULT
mt.ldap.security.password	The password to be used for LDAP authentication	NO VALUE SET BY DEFAULT

OIDC configuration properties

The following properties are required to configure WebSphere as an OIDC relying party for the Keycloak OIDC provider

Name	Description	Default
mt.oidc.enabled	If an OIDC provider is in place and should be configured for usage, please enable it with value 'true'. Any property with prefix `mt.oidc.` is only used if OIDC provider should be configured for usage as well.	`false`
mt.oidc.realmName	OIDC realm name	`connmt`
mt.oidc.identifier	OIDC identifier - the identifier for OIDC provider (keycloak)	`keycloak`
mt.oidc.clientId	OIDC client_id from OIDC provider	`connmt`
mt.oidc.clientSecret	OIDC client_secret from OIDC provider when registering Connections server as a client	NO VALUE SET BY DEFAULT

mt.oidc.rte.clientId	OIDC client_id for Rich Text Editor OAuth endpoint from OIDC provider	`conn-rte`
mt.oidc.rte.clientSecret	OIDC client_secret for Rich Text Editor OAuth endpoint from OIDC provider	NO VALUE SET BY DEFAULT
mt.oidc.ee.clientId	OIDC client_id for Embedded Experience OAuth endpoint from OIDC provider	`conn-ee-kc`
mt.oidc.ee.clientSecret	OIDC client_secret for Embedded Experience OAuth endpoint from OIDC provider	NO VALUE SET BY DEFAULT
mt.oidc.scope	OIDC scope of the token	`openid`
mt.oidc.host	OIDC hostname (without protocol or port). E.g. `oidc-provider.mycompany.com`	NO VALUE SET BY DEFAULT
mt.oidc.port	OIDC server port	`443`
mt.oidc.discoveryEndpointUrl	Discovery URL for OIDC provider endpoints. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/.well-known/openid-configuration`	NO VALUE SET BY DEFAULT
mt.oidc.authorizeEndpointUrl	Authorization URL for OIDC provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/protocol/openid-connect/auth`	NO VALUE SET BY DEFAULT
mt.oidc.tokenEndpointUrl	Token URL for OIDC provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/protocol/openid-connect/token`	NO VALUE SET BY DEFAULT
mt.oidc.logoutEndpointUrl	Logout URL for OIDC provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/protocol/openid-connect/logout`	NO VALUE SET BY DEFAULT
mt.oidc.introspectEndpointUrl	Introspect URL for OIDC provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/protocol/openid-connect/token/introspection`	NO VALUE SET BY DEFAULT
mt.oidc.jwkEndpointUrl	Certicate location URL on oidc provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt/protocol/openid-connect/certs`	NO VALUE SET BY DEFAULT
mt.oidc.issuerIdentifier	URL for issuing OIDC provider. E.g. `https://oidc-provider.mycompany.com/auth/realms/connmt`	NO VALUE SET BY DEFAULT
mt.oidc.signVerifyAlias	OIDC certificate name of server hosting keycloak. E.g. `oidc-provider.mycompany.com`	NO VALUE SET BY DEFAULT

MT provisioning configuration properties

The following properties are used to entitle existing users and organizations for using Connections MT

Name	Description	Default
provision.mt.url	The base Connections URL. E.g. `https://mymachine.mycompany.com:9443` or `https://mymachine.mycompany.com`	NO VALUE SET BY DEFAULT
provision.mt.ldapurl	The LDAP URL. E.g. `ldap://connmt-ldap.mycompany.com:389`	NO VALUE SET BY DEFAULT
provision.mt.ldapUserName	The LDAP user name / bind DN.	`cn=root`
provision.mt.ldapPassword	The LDAP password of the bind DN user	`password`
provision.mt.saasOrganizationsLdapDN	The Base DN of saasOrganizations, e.g. `cn=Organizations,ou=saasOrganizations,dc=mycompany,DC=com`	NO VALUE SET BY DEFAULT
provision.mt.defaultSubscriberLocale	Default language locale for users/subscribers	`en_US`
provision.mt.bssAdminUsername	Username of admin user that should be used for provisioning entitlements	`bssAdmin`
provision.mt.bssAdminPassword	Password of admin user that should be used for provisioning entitlements	`password`
provision.mt.saasOrganizationId	LDAP identifier attribute for organizations	`ibm-socialOrganizationId`

provision.mt.saasSubscriptionsRDN	LDAP relative DN for users	`cn=Users`
provision.mt.saasSubscriptionId	LDAP identifier attribute for users	`ibm-socialPersonId`
provision.mt.defaultQuotaSiz	Default file library size for users	`524288000`
provision.mt.defaultTransferQuota	Default transfer size for users	`102410000`
provision.mt.defaultAllowDataOverage	Defines whether users can leverage data overage	`true`
provision.mt.ldapOrganizationFilter	Filter for search queries over all organizations	`o=*`
provision.mt.ldapSubscriptionFilter	Filter for search queries over all subscriptions	`cn=*`

Running the auto deploy scripts

Prior to provisioning and entitling new users, there are just a few things to check after running the MT update scripts to validate that they have run properly.

1. Check all logs from the MT scripts for errors or warnings.
2. Verify that all WebSphere Nodes and Servers are running
3. Check the WebSphere logs for errors.

Once you have provisioned a user, login with that user to verify the that authentication with Keycloak works properly and that the various Connections apps are loading properly. Ensure that you create a community and prove that the Rich Content widget is loading properly (this widget uses an OAuth flow with Keycloak).

Post installation steps

Complete the following manual procedures after validating the auto deploy scripts.

1. Configuring the notification_v2 templates

To use Connections notification_v2 templates, the default templates for email notifications, follow these steps:

1. Enable all mail channels in notification-config.xml, as documented in Enabling email notifications on the Connections product documentation site.
2. Locate LotusConnections-config.xml in the deployment manager, add the following two properties in the <properties> section:

   ```
   <genericProperty name="mt_internalhostname">mtdemo1.cnx.cwp.pnp-
   hcl.com</genericProperty>
   <genericProperty name="mt_externalhostname">{org}.cnx.cwp.pnp-
   hcl.com</genericProperty>
   ```

   where:
   property: mt_internalhostname is the hostname from the URLs of the **<sloc:static>** attribute for any Connections service in the same file
   property: mt_externalhostname is the hostname from the URLs of the **<sloc:pattern>** attribute for any Connections service, which should have the {org}.xxx.com pattern in it
3. Sync the nodes, and restart all servers.

2. Configuring session management for the Blogs server

If Blogs is installed and run on its own cluster or server on a multiple-node deployment, follow these steps to avoid issues with creating new blog entries:

1. Follow the steps in this Support documentation to add the customer property HttpSessionIdReuse: https://www.ibm.com/support/pages/node/34099
2. On all nodes, use a session cookie name specifically for the Blogs server, for example, JSESSIONID_BLOGS. On the WAS admin console, use the following path to configure the session cookie name:

   Application servers > BlogsCluster_server1 > Session management > Cookies

Configure OrientMe web client container for Multi-Tenant

1. Setting environment variables for OrientMe

Within Component Pack, add two environment variables to the orient-web-client deployment.

Variables to add to the deployment :

```
- name: ORIENT_CNX_USE_REL_PATH
      value: "true"
- name: USE_REQUEST_HOST_FOR_CNX
      value: "true"
```

Edit the orient-web-client deployment in the editor add the env variables above using command:

```
kubectl -n connections edit deployment orient-web-client
```

Note: Caution on syntax spacing.

2. Setting JWT and LTPA token timeout to the same value

On kubernetes, please ensure that the configmap - connections-env variable jwt-expires-in-minutes is set to the same value as the LTPA timeout on WebSphere.

On WebSphere Admin console, you can find the LTPA timeout value by navigating:

```
Global security > LTPA
```

You can find the current value in kubernetes using command:

```
kubectl -n connections get configmap connections-env -o yaml | grep jwt-expires
```

You can use the 'edit' to modify the value using command:

```
kubectl edit configmap/connections-env -n connections
```

3. Add the pattern configuration to LotusConnections-config.xml for the orient service

- Edit the LotusConnections-config.xml from your WebSphere DMgr path. For example, /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/mtdemo1Cell01/LotusConnections-config/LotusConnections-config.xml

- Search for the section where serviceName="orient"

- Within the <sloc:href> block, add the following line, but replace *connections_domain.com* with your actual domain.

```
 <sloc:pattern href="http://{org}.connections_domain.com"
ssl_href="https://{org}.connections_domain.com"/>
```

- This section should now look similar to this:

```
  <sloc:serviceReference bootstrapHost="admin_replace"
bootstrapPort="admin_replace" clusterName="" enabled="true" serviceName="orient"
ssl_enabled="true">

    <sloc:href>
```

```
      <sloc:hrefPathPrefix>/social</sloc:hrefPathPrefix>
      <sloc:static href="http://cnxserver.connections_domain.com"
ssl_href="https://cnxserver.connections_domain.com"/>
      <sloc:pattern href="http://{org}.connections_domain.com"
ssl_href="https://{org}.connections_domain.com"/>
      <sloc:interService href="https://cnxserver.connections_domain.com"/>
    </sloc:href>
  </sloc:serviceReference>
```

- Save the file, do a full synchronization of Nodes using WebSphere administration and restart the server.