Single node installation on Linux - CustomerID

Last reviewed: 2018-05-04

Make sure you meet the Installation requirements first.

Follow the steps in order. Issue all commands using the **root** user account.

Perform on Ubisecure CustomerID server (or relating to it):

1. **Back up Ubisecure Directory.** See the instructions in Backup and restore Ubisecure Directory - SSO.
2. **Unpack the distribution package.**
   Extract the Ubisecure CustomerID distribution package in your home folder:

   ```
   mkdir ~/customerid
   cp customerid-x.x.x-linux.tar.gz ~/customerid/
   cd ~/customerid
   tar xzvf customerid-x.x.x-linux.tar.gz
   ```

3. **Check Java.**

   Ubisecure CustomerID requires the correct versions of:

   - Java Runtime Environment (JRE) for Servers
   - The Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files to be included in the Java installation

   Please check the Java installation based on requirements mentioned in Installation and (upgrade) requirements - CustomerID.

   If you haven't done so already during Java installation, make sure you have the following environment variables set related to Java. The values below are just examples; modify the paths according to your Java installation.
   • Set the **JAVA_HOME** environment variable to JDK folder (eg. `/usr/local/java/jdk1.8.0_181`)
   • Set the **JRE_HOME** environment variable to JRE folder (eg. `/usr/local/java/jdk1.8.0_181/jre`)

   ### Additional SSL Considerations

   Ubisecure CustomerID can be configured to make calls to third-party software during the user registration workflows. Typically, data entered by the user is verified against a CRM or other backend service to determine which access rights a user should be given automatically based on an existing service agreement.

   If you plan to use back channel connections from Ubisecure CustomerID over SSL encrypted connections, you will have to add each server's public key to the Server JRE's cacerts file. You can find the cacerts file under `${JRE_HOME}/lib/security/cacerts`. Once you have downloaded the server's public key, you can add it to the key store with the following commands:

   ```
   cd ${JRE_HOME}/lib/security
   ${JAVA_HOME}/bin/keytool –importcert –trustcacerts –alias "<descriptive alias here>" .keystore cacerts –
   storepass changeit –file C:\path\to\certificate.cer
   ```

   These commands can also be run at a later stage when third-party backend services are added to user registration workflows.

4. **Install WildFly.** See the instructions in WildFly installation on Linux - CustomerID.
5. **Extract the deployment template.**
   Create the folder `/usr/local/ubisecure` if one does not exist yet and extract the archive `cid-deployment-template-x.x.x-linux.tar.gz` therein:

   ```
   mkdir -p /usr/local/ubisecure
   cd /usr/local/ubisecure
   tar xzvf ~/customerid/cid-deployment-template-x.x.x-linux.tar.gz
   ```

   An optional additional step is to also copy the file containing versioning information from the installation package to the installation folder:

   ```
   cp ~/customerid/customerid-x.x.x-versioninfo.txt /usr/local/ubisecure/
   ```

6. **Edit the setup template and run setup.** See the instructions in Setup template on Linux - CustomerID.
7. **Configure WildFly.** See the instructions in WildFly configuration on Linux - CustomerID.

8. **Prepare PostgreSQL.** See the instructions in PostgreSQL preparation on Linux - CustomerID.
9. **Create a JDBC data source to WildFly.**
   Ubisecure CustomerID uses a JDBC data source to access the database, thus one needs to be created to WildFly before the Ubisecure CustomerID application can be deployed. There is a script in the distribution package's tools folder for this purpose: `create-datasource.sh`. Note that the `linux.config` file must have been configured, `setup.sh` must have been run successfully, and WildFly must be running before the `create-datasource.sh` script can be run successfully.

   ```
   cd /usr/local/ubisecure/customerid/tools
   ./create-datasource.sh
   ```

10. **Create a directory service for** Ubisecure CustomerID **SQL** in SSO Management. See the instructions in SQL directory service creation on Linux - CustomerID.
11. **Create web agents for Ubisecure CustomerID.**
    Ubisecure CustomerID needs two web agents. The first one is used to provide login functionality to the Ubisecure CustomerID user interfaces and also the LDAP user account that Ubisecure CustomerID uses when accessing Ubisecure Directory. The second web agent is used when performing verifications during registrations. Ubisecure CustomerID installation package contains LDIF import files that need to be imported to Ubisecure Directory using the import functionality of Ubisecure SSO.

    ### Importing the web agents:

    1. Copy the LDIF files in `/usr/local/ubisecure/customerid/application/ldap` on the Ubisecure CustomerID server to the Ubisecure SSO server. You can place them, for example, in the home directory in a folder called `customerid-ldifs`.
    2. Use the script `import.sh` in the path `UBILOGIN_HOME/ldap/openldap/import.sh` to import these files:

    ```
    cd /usr/local/ubisecure/ubilogin-sso/ubilogin/ldap/openldap
    ./import.sh ~/customerid-ldifs/customerid.ldif
    ./import.sh ~/customerid-ldifs/customerid-secrets.ldif
    ```

    ⚠ **NOTE:** If the import script prompts for the LDAP password, you can find the correct password in the file `/usr/local/ubisecure/ubilogin-sso/ubilogin/unix.config` in the property `openldap.root.password`.

12. **Create a directory service for** Ubisecure CustomerID **LDAP** in SSO Management. See the instructions in LDAP directory service creation on Linux - CustomerID.

Perform on each Ubisecure SSO node:

1. **Install PostgreSQL JDBC driver to the SSO node(s).**
   Ubisecure CustomerID package includes the PostgreSQL JDBC driver.

   ⚠ **NOTE:** The installation instructions concerning PostgreSQL JDBC driver to SSO are written for a single Ubisecure SSO node. If you have more nodes, these instructions should be followed on all of them.

   ### To install the PostgreSQL JDBC driver to Ubisecure SSO:

   Transfer the library `postgresql-x.x.x.jar` to the Ubisecure SSO server and copy it to the folder `$JRE_HOME/lib/ext`.

2. **Install Ubisecure CustomerID SSO Adapter to the SSO node(s).** See the instructions in SSO adapter installation on Linux - CustomerID.

Perform on Ubisecure CustomerID server (or relating to it):

1. **Add authentication method configurations** in Ubisecure SSO Management. See the instructions from Authentication method configuration on Linux - CustomerID.
2. **Create a site specific configuration** for Ubisecure CustomerID.

   ⚠ **NOTE:** This step is very important as some configuration options cannot be changed after this step.

   Creating a site specific configuration for Ubisecure CustomerID can be done by editing the file `custom/eidm2.properties` and other Ubisecure CustomerID configuration files. For more information about the configuration options, refer to *Configuration - CustomerID*.

   Examples of the types of configurations required include:

   - defining user registration workflows
   - defining organization types and roles
   - defining what strong authentication methods are available
   - defining policies for login names

Generally, it is recommended to use a very basic `eidm2.properties` configuration first, ensure the system is fully configured and running, and then modify the settings again later to match the use case requirements.

Execute the following commands to create and edit the `eidm2.properties` file:

```
cd /usr/local/ubisecure/customerid/application/custom
nano eidm2.properties
```

Typical entries include:

```
# use email address as the username when logging in
# this requires that directory.account.login=mail is added to password.2 authentication method settings
# without this setting, the default uid is used as the username when logging in
general.login.attribute = mail

# where to redirect the user when an error occurs or user presses exit - generally home page of the
target service
general.default.returnUrl = https://www.example.com

# where to redirect the user after logout has been performed
general.default.logoutReturnUrl = https://www.example.com
```

3. **Download Identity Provider metadata** from Ubisecure SSO **and generate Service Provider metadata**

   1. Download IDP metadata by running the following commands:

   ```
   cd /usr/local/ubisecure/customerid/tools
   ./get-metadata.sh
   ```

   2. Initialize the Ubisecure CustomerID SPs:

   ```
   cd /usr/local/ubisecure/customerid/tools
   ./init-eidm-sp.sh
   ```

   3. Initialize the authentication provider:

   ```
   cd /usr/local/ubisecure/customerid/tools
   ./init-eidm-ap.sh
   ```

4. **Deploy Ubisecure CustomerID to WildFly.**

   Ubisecure CustomerID uses WildFly as a J2EE Container. The next step is to deploy the `cid-ear-x.x.x.ear` and `cid-worker-ear-x.x.x.ear` enterprise archives (EARs).

   Deploy the Ubisecure CustomerID applications to WildFly using the `deploy-ear.sh` command. When invoking the command, you must supply the path to the ear file, like in the example below:

   ```
   cd /usr/local/ubisecure/customerid/tools
   ./deploy-ear.sh ~/customerid/cid-ear-x.x.x.ear
   ./deploy-ear.sh ~/customerid/cid-worker-ear-x.x.x.ear
   ```

5. **Configure SELinux.**
   If a reverse proxy is used in SELinux:

   ```
   /usr/sbin/setsebool httpd_can_network_connect 1
   ```

6. **Restart Ubisecure CustomerID.**

   When we restart Ubisecure CustomerID, we will also set final permissions to the installation folder for the user **wildfly** to make sure we won't have problems with file permissions.

```
systemctl stop wildfly.service
chown -R wildfly. /usr/local/ubisecure/customerid
chmod -R o-xrw /usr/local/ubisecure/customerid
systemctl start wildfly.service
```

7. **Initialize data storages.**
   Initialize database and repository by running the following scripts:

```
cd /usr/local/ubisecure/customerid/tools
./init-customerid-data-storages.sh
./get-metadata-for-ap.sh
```

Perform on each Ubisecure SSO node:

Restart Ubisecure SSO.

Run the following commands:

```
/etc/init.d/ubilogin-server stop
/etc/init.d/ubilogin-server start
```

Perform on Ubisecure CustomerID server (or relating to it):

1. **Restart Ubisecure CustomerID.**

```
systemctl stop wildfly.service
chown -R wildfly. /usr/local/ubisecure/customerid
chmod -R o-xrw /usr/local/ubisecure/customerid
systemctl start wildfly.service
```

2. **Import the example admin user.**
   After installing the software, it is necessary to create an administrative user. It is recommended that generic administrative accounts are not used.

To import the user organization and the first user account:

In the folder `/usr/local/ubisecure/customerid/tools`, modify the provided template import file:

```
cd /usr/local/ubisecure/customerid/tools
nano examples/importtool/example.import
```

Include your personal account. Then execute the import:

```
./import.sh examples/importtool/example.import
```

For more details, refer to the page *Data import from external systems - CustomerID*.