

Add-ons

Introduction

Add-ons for HUBzero are available here. Currently these consist of a couple projects that have not yet been fully integrated into the the HUBzero packaging and installation process.

[Hubgraph](#) - Alternate Search Engine

[Shibboleth/InCommon](#) - Authentication Plug-In

Hubgraph search

Prerequisites

Hubgraph is a node.js module, requiring an installation of node.js of at least the 0.10.x branch, and npm, the node package manager.

If your distribution does not package an appropriate version of node, see [the node.js downloads page](#)

The current reference version of node.js is 0.10.29, the latest version packaged by Debian as of this writing.

There are also build dependencies. npm is a source-based package manager. These packages are:

libboost-filesystem-dev libboost-regex-dev libboost-serialization-dev libboost-graph-dev libboost-system-dev libicu-dev

These should be available through your distribution's package manager.

Installation

The recommended way to install hubgraph is with npm through its global repository:

```
npm install hubgraph-hubzero
```

This installs a few packages we maintain, with git repositories listed here:

hubgraph-hubero	https://bitbucket.org/snyder13/hubgraph-hubzero
hubgraph-ext	https://bitbucket.org/snyder13/hubgraph-ext
hubgraph	https://bitbucket.org/snyder13/hubgraph

Installing from git is not recommended. The latest `master` copy of each branch is not guaranteed to be mutually compatible with the other branches. The npm modules are synced with each other to guarantee a common vintage.

If you do install from git, npm is still required: cd to the `source` directory and run `npm install .`

Configuration

ADD-ONS

Step-by-step instructions for configuring hubgraph are available through the package:

```
node -e 'require("hubgraph-hubzero").install()'
```

A few notes:

- `node` may be called `nodejs`, particularly on Debian installations
- You might need to supply the path where `npm` installed hubgraph if it happens to be different from the path your `node` executable looks, eg:
NODE_PATH=/usr/local/lib/node_modules nodejs -e 'require("hubgraph-hubzero").install()'

Running

```
node -e "require('hubgraph-hubzero').server()"
```

Help?

If you have any problems please visit <https://hubzero.org/support> and submit a support ticket describing the situation. Please include specific details about your Linux distribution and your version of node.js and npm.

Shibboleth authentication

Introduction

This plugin provides some code necessary to allow your hub to accept credentials using the Shibboleth system. Most commonly, this implies membership in the InCommon network.

Shibboleth has some particular architectural demands, namely that it will install a new daemon and a new Apache module on your system. InCommon has some administrative demands, in that you will need to negotiate to get your hub added to their XML manifest as a service provider.

If those are steps you're prepared to take, carry on.

Download

- [git repository](#)
- [source tarball](#)
- [Debian Wheezy package](#) (architecture independent)
- [Redhat 6.x package](#) (architecture independent) (note dependencies section)

Dependencies

If you installed the hubzero-shibboleth package on Debian, you're set. The relevant packages were included as dependencies. The packages are:

```
shibboleth2-sp-utils shibboleth2-sp-schemas libapache2-mod-shib2
```

At the time of this writing, Shibboleth is not distributed in the core repositories for Redhat/CentOS. You can read about how to add a repo that has what you need here:

<https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPLinuxRPMInstall>

Generate a private key

Use shib-keygen to generate /etc/shibboleth/sp-key.pem. Note that this utility may not be on your path unless you are root.

Configure Shibboleth

[Shibboleth official quickstart documentation reference](#)

The main configuration file is located at `/etc/shibboleth/shibboleth2.xml`. There are some other files that might be of interest to you here, but the defaults are acceptable to get your hub working with InCommon.

In `shibboleth2.xml`:

- Update `<ApplicationDefaults entityID="{url}" ...>` so `{url}` is `https://{your hostname}/Shibboleth.sso`. This is your Shibboleth endpoint, designated later by the Apache configuration as the location where the shib2 module will manage communication with ID providers.
- Update `<Sessions ...handlerSSL="false" ...>` to `handlerSSL="true"`, if it is not already

Configure Apache

[Shibboleth official Apache configuration reference](#)

Ensure that Apache is loading the module. Typically this means that there is a link in `mods-enabled` to `shib2.load` in `mods-available`

In `-s /etc/apache2/mods-available/shib2.load /etc/apache2/mods-enabled`

If you do not have this directory structure you can also enable the module directly in the next step by adding this to your Apache configuration file:

```
LoadModule mod_shib /usr/lib/apache2/modules/mod_shib2.so
```

In the conf file defining your SSL host, (usually located in `/etc/apache2/sites-enabled`):

- If not already set in the SSL `<VirtualHost>` `UseCanonicalName on`;
- To enable shibd's endpoint, add: `<Location /Shibboleth.sso> SetHandler shib </Location>`
- Joomla! routing will stomp on `/Shibboleth.sso` unless you change the `mod_rewrite` rules a bit.
 - You should have a line like: `RewriteRule (.*?) index.php` probably preceded by a few `'RewriteCond'`'s. Add a new condition to exempt the shib2-controlled path:

```
RewriteCond %{REQUEST_URI} !/Shibboleth.sso/.*$ [NC]
```

Restart apache: `/etc/init.d/apache2 restart`

Verify

From the same host (this is IP-restricted):

```
wget -q --no-check-certificate https://localhost/Shibboleth.sso/Metadata -O - | tee /etc/shibboleth/sp-metadata.xml
```

This command should write XML to the listed file (and stdout) wrapped in `<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" ...>`

If it does not, review the references above to troubleshoot.

You may skip to "Configuring Joomla!" if you do not want to test interop more thoroughly with the TestShib ID provider, but I would recommend you do this test.

Upload metadata to TestShib.org

- Copy the metadata generated above to some unique name, for example:

```
cp /etc/shibboleth/sp-metadata ~/ {your hostname}-sp-metadata.xml
```

- Upload that file here: <https://www.testshib.org/register.html>. Uploading a file of the same name will overwrite it on the testshib server, should you need to make any adjustments.

Change your local configuration to accept TestShib as an ID provider

- Visit this URL to get an appropriate test configuration XML file:

```
https://www.testshib.org/cgi-bin/sp2config.cgi?dist=Others&hostname={your hostname from the Shibboleth Configuration step above})
```

- Assuming that looks OK, copy the output over your existing `/etc/shibboleth/shibboleth2.xml`:

```
wget -q --no-check-certificate "https://www.testshib.org/cgi-bin/sp2config.cgi?dist=Others&hostname={your hostname}" -O /etc/shibboleth/shibboleth2.xml
```

Restart services: `/etc/init.d/shibd restart && /etc/init.d/apache2 restart`

Configuring Joomla!

If you do not already have `plg_authentication_shibboleth` installed, this package installs a tarball in `$(PREFIX)/usr/lib/hubzero` that you may install using Joomla!'s package management interface at `/administrator`.

If you are building from git, running `make` in the source directory will build this file.

Manage ID providers on Joomla!'s admin page

In Extensions->Plugins, select Authentication - Shibboleth

Ideally it should look a lot like the screenshot in that it found `testshib` in your XML configuration. If so, you can click the down arrow by that entry to move it into your active provider list.

This may fail if, for example, `shibboleth2.xml` is not readable by the web user, or if you changed your configuration so that the file is located somewhere unexpected.

It is not necessary, however, for the web server to read this file. If you'd like you can simply enter the EntityID for `testshib` (<https://idp.testshib.org/idp/shibboleth>) in the white box with the button labeled "Add ID provider". Enter something, eg "TestShib" for the label.

Quick run-down of the fields here:

- Entity id: (required) corresponds to the corresponding `entityId` in `shibboleth2.xml` and must match exactly for things to work out.
- Label: (required) name to show on the log-in button of your hub for this provider
- Initialism: (optional) if you have more than ten supported ID providers, the log-in list becomes searchable, and in this case you can add a short name for institutions so that they will come up when the user types that as well as when they type a portion of the label. (For example, if you federated with the National

Science Foundation you might add "NSF" here)

- Host: (optional) institutions may be pre-selected if the IP address of the user looks like it is in a particular network, eg, to follow the previous example, nsf.gov to pre-select the National Science Foundation
- Logo: (optional) also shown on the button. Enter a URL here to make a iconified copy of it. You may have better results in some cases if you resize to no more than 28px in either extent yourself.

Finally, you can select the order in which you would like the button to appear on the login page here. When you're done, click "Save & Close" in the top right. This will take you back to the screen where you can click the icon in the "Status" column to enable the plug-in.

Try logging in!

(If you run into any problems here, there might be a clue in [TestShib's logs of its ID provider actions](#))

Since TestShib doesn't release any attributes, you'll have to enter a name when you log in. Hopefully you can negotiate to get names and emails released to your hub with "real" ID providers, which you're now clear to do if everything worked out.

Help?

- If you have a problem that you can't resolve that appears to be related to Shibboleth's machinery, please consult [the official documentation](#) carefully.
- If you can't resolve the problem, there is a mailing list: <https://shibboleth.net/community/lists.html>
-

If the problem you are experiencing appears to be related not to the Shibboleth interchange mechanism but to something in the hub's implementation of the log-in procedure, visit <https://hubzero.org/support> to enter a support ticket describing the situation.

Separate execution host setup

Intro

Our standard open source install sets up a web server and execution host on a single machine. For smaller hubs, this is an adequate setup, however on hubs with greater needs, often separate (and sometimes multiple) execution hosts are required for an installation. The following is a set of directions for setting up an execution host.

Note, these directions are not complete step by step directions, but more of a guideline for setting up an execution host. A user should have a very thorough understanding of hub architecture before attempting to setup an additional execution host.

Setup steps

1. Do standard debian OS install
2. Configure hostname and /etc/sources.list appropriately
3. Setup standard hub through the openldap step
4. Install openvz kernel
5. Install hubzero mw-service on execution host 'apt-get install -y hubzero-mw-service'
6. Run 'mkvztemplate amd64 wheezy diego'
7. run 'hzcms configure mw-service --enable'
8. configure /etc/nslcd.conf and restart. /etc/nslcd.conf will need the following modifications:
 - URI - modified to point to the ldap on the web host
 - binddn - set to the search user dn on the webserver (do a 'slapcat | grep search' to get the DN for the search user on the web server)
 - bindpw - set to the value contained for the LDAP-SEARCHPW in the /etc/hubzero.secrets file on the web server
9. Install hubzero mw-client and configure on execution host
10. copy /etc/mw-client/maxwell.key.pub from web host to /root/.ssh/authorized keys file on execution host
11. On web server, add execution host to tools component
 - login to webserver admin section (webserver/administrator)
 - select components->tools
 - on host tab click on + sign in upper right to add an execution host
 - When you are returned to the list of hosts, you should see two, one for the web server, likely called localhost and the IP for your execution host
 - Under the provisions section, click on pubnet, sessions, and workspace for the new execution host
 - Under the provisions section, uncheck everything but fileserver for your web server
12. Setup nfs server on web server
 - 'apt-get install nfs-kernel-server'

edit /etc/exports to export /home and /apps, something similar to this:

```
/home executionhost.ip.address(rw,no_subtree_check)
```

```
/apps executionhost.ip.address(rw,no_subtree_check)
```

13. Setup nfs client on execution host

```
apt-get install nfs-common
```

```
mount -t nfs webserver:/home /home
```

```
mount -t nfs webserver:/apps /apps
```

NOTE: user will want to add appropriate sections in the /etc/fstab file to remount these locations after a reboot. Something similar to:

```
your.webserver.org:/apps /apps nfs vers=3,rw 0 0
```

```
your.webserver.org:/home /home nfs vers=3,rw 0 0
```