

Installation and Usage of CLASSE Federation and VO components for Administrators – Juno Version

Date	Author	Notes
15/09/14	Kristy Siu	First Version
15/09/14	Kristy Siu	Removed references to VO for later document
28/11/14	Ioram Sette	Major review, added VO references
02/12/14	Ioram Sette	Structural changes
04/12/14	Ioram Sette	Review after testing
11/12/14	Ioram Sette	Added a step to add a new protocol (links an IdP to a mapping entry)
22/12/14	Alejandro Perez Mendez	Added instructions to install the required version of python-pyscss library.
13/01/15	Ioram Sette	Support to Abfab protocol
31/01/15	Ioram Sette	David Review. Improved Abfab protocol instructions.
02/02/15	Ioram Sette	Updated for Ubuntu 12.04
04/02/15	Ioram Sette	Added more details about Abfab. Added “how to setup federated administrators”. Replaced loggedId attribute by REMOTE_USER.
5/2/15	David Chadwick	Minor edits and comments
5/2/15	Ioram Sette	Completed after David Review.
17/2/15	Ioram Sette	Modified and added a new subsection describing how to set privileges to a VO Role via an Openstack Role.
20/2/15	Ioram Sette	Updated with Alejandro’s reviews. Added a new permission required by VoAdmins due to improvements in code.
23/2/15	Ioram Sette	Update due to changes in the code.
23/2/15	David Chadwick	Changes to grammar

Table of Contents

<u>1</u>	<u>INTRODUCTION AND OVERVIEW</u>	<u>3</u>
<u>2</u>	<u>INSTALLATION AND CONFIGURATION OF OPENSTACK WITH SHIBBOLETH IDP</u>	<u>4</u>
2.1	REQUIREMENTS	4
2.2	KEYSTONE	5
2.2.1	DOWNLOADING AND INSTALLING THE SOFTWARE	5
2.2.2	CONFIGURING MYSQL	5
2.2.3	CONFIGURING RABBITMQ	5
2.2.4	CONFIGURING KEYSTONE	5
2.2.5	POPULATE DATABASE – PART 1	7
2.2.6	CONFIGURING APACHE2	7
2.2.7	CONFIGURING MOD_SHIB	10
2.2.8	POPULATE DATABASE – PART 2	10
2.3	HORIZON	11
2.3.1	DOWNLOADING AND INSTALLING THE SOFTWARE	11
2.3.2	CONFIGURING APACHE2	12
2.4	DJANGO_OPENSTACK_AUTH	13
2.5	KEYSTONECLIENT	13
2.6	CONFIGURING FEDERATION	14
<u>3</u>	<u>CONFIGURATION OF OPENSTACK WITH ABFAB IDPS</u>	<u>16</u>
3.1	REQUIREMENTS	16
3.1.1	DJANGO OPENSTACK AUTH – CHANGES IN THE CODE	16
3.1.2	INSTALL MOONSHOT LIBRARIES ON THE OPENSTACK SERVER	16
3.1.3	INSTALL AND CONFIGURE A MOONSHOT IDP	17
3.2	CONFIGURATION OF THE MOONSHOT PLUGIN	18
3.3	CONFIGURATION IN APACHE FOR THE MOONSHOT PLUGIN	18
3.4	CONFIGURING ABFAB IN KEYSTONE	19
3.5	MOONSHOT SSP	21
<u>4</u>	<u>CONFIGURING OPENSTACK PRIVILEGES TO A VO ROLE</u>	<u>22</u>
4.1	ASSIGN OPENSTACK ROLES TO VO ROLES	22
4.2	ASSIGN AUTHORISATION POLICIES TO OPENSTACK ROLES	23
<u>5</u>	<u>USAGE</u>	<u>25</u>
5.1	FEDERATED HORIZON	25
5.1.1	LOGGING IN	25
5.1.2	IDENTITY PROVIDER MANAGEMENT	25
5.1.3	MAPPING MANAGEMENT	29
<u>6</u>	<u>USEFUL LINKS</u>	<u>31</u>

1 Introduction and Overview

This guide aims to facilitate the installation and usage of the components produced during the CLASSe project. The project modified the existing OpenStack services, Keystone and Horizon to allow web based federated authentication and VO management. It also provided additional interfaces to some existing Keystone operations. These were: federated authentication; identity provider management, attribute mapping management, vo management, vo role management. The guide will provide instructions for downloading, installing and configuring each modified component. Following this, a usage guide for the new Horizon interfaces will be given.

Section 2 presents the installation and configuration guide for Openstack using Shibboleth IdPs. Section 3 shows how to disable Shibboleth and to enable Abfab IdPs. Usage presented in section 4 is common to both federation protocols. Section 5 presents useful links.

IMPORTANT NOTE: INSTALLATION ORDER

The order in which components are installed can affect the installation process. It is recommended that the given installation order in section 2 is followed to avoid any conflicts in the imported libraries.

2 Installation and Configuration of Openstack with Shibboleth IdP

The modified code has been tested to work on **Ubuntu 12.04.5** and **Ubuntu 14.04.1** and it is available for download from Github.com. See each component section below for the repository location. This manual will give instructions on how to download it using git.

2.1 Requirements

Install a fresh copy of **Ubuntu 12.04.5** or **Ubuntu 14.04.1** and update its modules using the following commands. Details about a specific distribution are marked in **red**.

```
$> sudo apt-get update
$> sudo apt-get dist-upgrade
```

The following packages should be installed before proceeding:

- python, python-dev, pip
- git
- apache2 (Keystone and Horizon run under Apache2)
 - mod_wsgi (Keystone's and Horizon's dependency)
 - mod_shib (Keystone's dependency for Federation)
- mysql (Keystone's dependency)
- rabbitmq (Keystone's dependency)
- lxml (only needed on Ubuntu 12.04)

```
$> sudo apt-get install python-pip
$> sudo apt-get install python2.7 python2.7-dev
$> sudo apt-get install git
$> sudo apt-get install apache2 libapache2-mod-wsgi
$> sudo apt-get install libapache2-mod-shib2
$> sudo apt-get install mysql-server mysql-client python-mysqldb
$> sudo apt-get install rabbitmq-server
$> sudo apt-get install python-lxml
```

Please Note

1. If you are not using Ubuntu 12.04.5 or 14.04, package names may vary. Please refer to documentation for your Operating system and equipment if you cannot find the packages listed.
2. During mysql installation, you will set up a password for the root user. Remember this password for later configuration.

An specific version of python-pyscss needs to be installed as well. In particular, the version from Ubuntu 14.10 (utopy) is required.

For doing that, on Ubuntu 14.04, you should (assuming an amd64 architecture):

```
$> wget http://archive.ubuntu.com/ubuntu/pool/main/p/python-pyscss/python-pyscss\_1.2.1-0ubuntu2\_amd64.deb
$> sudo dpkg -i python-pyscss_1.2.1-0ubuntu2_amd64.deb
```

On Ubuntu 12.04, do:

```
$ pip install pyscss==1.2.1
```

2.2 Keystone

2.2.1 Downloading and installing the software

The Keystone software can be obtained from the following GIT repository.

https://github.com/ioram7/keystone/tree/federated_vo_management

Create a directory to download Keystone and other required openstack packages (eg.: /usr/share/openstack) and move into it. We will use `OPENSTACK_HOME` to refer to this directory. Using `sudo` can be necessary since you need permission to create and write in this directory. Add “`sudo`” before the following commands when necessary.

```
$> mkdir OPENSTACK_HOME
$> cd OPENSTACK_HOME
```

Keystone can be downloaded using the following command.

```
$> git clone -b federated_vo_management
    https://github.com/ioram7/keystone.git
```

To install it you should run the following commands:

```
$> cd keystone
$> sudo pip install -r requirements.txt
$> sudo python setup.py install
```

2.2.2 Configuring MySQL

Run the commands below to setup and configure MySQL. Please provide MySQL root password when required.

```
$> sudo mysql_install_db
```

Create a database Keystone database, Keystone’s database user, define a password to it and set its privileges on keystone database.

```
$> sudo mysql -u root -p
mysql> CREATE DATABASE keystone;
mysql> GRANT ALL PRIVILEGES ON keystone.* to 'keystone'@'localhost'
    IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON keystone.* to 'keystone'@'%'
    IDENTIFIED BY 'password';
```

2.2.3 Configuring RabbitMQ

Run the following lines to configure RabbitMQ.

```
$> sudo rabbitmqctl add_user keystone password
$> sudo rabbitmqctl set_permissions -p "/" keystone ".*" ".*" ".*"
```

2.2.4 Configuring Keystone

Create a directory named `/etc/keystone` and copy the following files from `OPENSTACK_HOME/keystone/etc` into it. Remember to give reading permission to Apache’s user (eg. `www-data`) in this directory.

```
$> sudo mkdir /etc/keystone
$> sudo cp OPENSTACK_HOME/keystone/etc/keystone.conf.sample
    /etc/keystone/keystone.conf
```

```
$> sudo cp OPENSTACK_HOME/keystone/etc/keystone-paste.ini
    /etc/keystone
$> sudo cp OPENSTACK_HOME/keystone/etc/policy.json /etc/keystone
```

Edit the file `/etc/keystone/keystone.conf`, and add the following lines in the `[DEFAULT]` section. Parameters (highlighted in blue) must be modified according to your environment.

```
max_token_size = 16384
logging_exception_prefix = %(process)d TRACE %(name)s %(instance)s
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d
logging_default_format_string = %(process)d %(levelname)s %(name)s [-
] %(instance)s%(message)s
logging_context_format_string = %(process)d %(levelname)s %(name)s
[% (request_id)s %(user_identity)s] %(instance)s%(message)s
debug = True
admin_token = password
admin_bind_host = 129.12.3.224
admin_endpoint = http://service.kent.ac.uk:%(admin_port)s/
public_endpoint = http://service.kent.ac.uk:%(public_port)s/
rabbit_host = 129.12.3.224
rabbit_password = password
```

The parameter `admin_token` can have any value. It's recommended to configure it as a random value, like a hash. The parameter `rabbit_password` must contain the password set up in RabbitMQ configuration step.

Add the following line in the `[assignment]` section.

```
driver = keystone.assignment.backends.sql.Assignment
```

Add (or uncomment) the following line in the `[catalog]` section.

```
driver = keystone.catalog.backends.sql.Catalog
```

Add (or uncomment) the following line in the `[identity]` section.

```
driver = keystone.identity.backends.sql.Identity
```

Add the following line in the `[token]` section.

```
driver = keystone.token.backends.sql.Token
```

Add (or edit) the following lines in the `[auth]` section.

```
methods=external,password,token,saml2
saml2=keystone.auth.plugins.mapped.Mapped
```

Add (or uncomment) the following line in the `[federation]` section.

```
driver=keystone.contrib.federation.backends.sql.Federation
```

Add the following line in the `[database]` section. Please provide keystone's database username and password as defined.

```
connection=mysql://keystone:password@127.0.0.1/keystone?charset=utf8
```

Edit `/etc/keystone/keystone-paste.ini` and add the following lines, if they don't exist.

```
[filter:redirect_extension]
paste.filter_factory =
    keystone.middleware.HorizonRedirectMiddleware.factory

[filter:vo_extension]
```

```
paste.filter_factory =
    keystone.contrib.virtual_organisations.routers:
    VirtualOrganisationExtension.factory

[filter:federation_extension]
paste.filter_factory =
    keystone.contrib.federation.routers:FederationExtension.factory
```

In the same file, edit the [pipeline:api_v3] section according to the example:

```
[pipeline:api_v3]
pipeline = sizerlimit url_normalize build_auth_context token_auth
    admin_token_auth xml_body_v3 json_body redirect_extension
    ec2_extension_v3 s3_extension simple_cert_extension
    revoke_extension federation_extension vo_extension service_v3
```

Edit the **policy.json** and add the line at the end of the file. Don't forget to add a comma (,) before this line (at the end of the previous line).

```
'identity:vo_admin': "rule:admin_required"
```

Run the following command to setup keys for signing PKI tokens.

```
$> sudo keystone-manage pki_setup --keystone-user www-data
    --keystone-group www-data
```

Then, give permission to apache user read keystone configuration files.

```
$> sudo chown -R www-data.www-data /etc/keystone
```

Please Note

Additional documentation about installing and configuring a Keystone server can be found in the following sites.

<http://docs.openstack.org/developer/keystone/installing.html>
<http://docs.openstack.org/developer/keystone/configuration.html>

2.2.5 Populate Database – Part 1

The following lines create Keystone tables and populate them, including Federation and VO ones.

```
$> sudo /bin/sh -c "keystone-manage db_sync" keystone
$> sudo keystone-manage db_sync --extension federation
$> sudo keystone-manage db_sync --extension virtual_organisations
```

2.2.6 Configuring Apache2

In order for federation to be used with Keystone it is required that a Apache2 HTTP server using either mod_shib or mod_mellon protects your Keystone server endpoint. This guide will present instruction to configure mod_shib.



Restart Apache2 after making any configuration change!

```
sudo service apache2 restart
```

2.2.6.1 Install Keystone's Scripts

Create a directory for Keystone inside Apache's public folder, for instance: `"/var/www/cgi-bin/keystone"` and link Keystone's executable into it.

```
$> sudo mkdir -p /var/www/cgi-bin/keystone
$> cd /var/www/cgi-bin/keystone
$> sudo ln -s OPENSTACK_HOME/keystone/httpd/keystone.py main
$> sudo ln -s OPENSTACK_HOME/keystone/httpd/keystone.py admin
$> sudo chown -R www-data.www-data /var/www/cgi-bin/keystone
```

2.2.6.2 Configure Keystone's Site

Enable WSGI and `mod_shib` modules.

```
$> sudo a2enmod wsgi
$> sudo a2enmod shib2
```

Create Keystone's site configuration file (**keystone.conf**) into `/etc/apache2/sites-available`, according to the example below.

```
Listen 5000
Listen 35357

# Standard Keystone endpoint
<VirtualHost *:5000>

    # Set the location of the shibboleth endpoint
    <Location /Shibboleth.sso>
        SetHandler shib
    </Location>

    # Protect the Keystone federated endpoint for mod_shib
    <LocationMatch /v3/OS-
FEDERATION/identity_providers/Shib.*?/protocols/saml2/auth>
        ShibRequestSetting requireSession 1
        AuthType shibboleth
        ShibRequireSession On
        ShibExportAssertion Off
        Require valid-user
    </LocationMatch>
    WSGIDaemonProcess keystone-public processes=5 threads=1 user=www-
data display-name=%{GROUP}
    WSGIProcessGroup keystone-public
    WSGIScriptAlias / /var/www/cgi-bin/keystone/main
    WSGIApplicationGroup %{GLOBAL}
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>

# Admin Keystone endpoint
<VirtualHost *:35357>
    # Set the location of the shibboleth endpoint
    <Location /Shibboleth.sso>
        SetHandler shib
```



```

</Location>

# Protect the Keystone federated endpoint for saml2
<LocationMatch /v3/OS-
FEDERATION/identity_providers/Shib.*?/protocols/saml2/auth>
    ShibRequestSetting requireSession 1
    AuthType shibboleth
    ShibRequireSession On
    ShibExportAssertion Off
    Require valid-user
</LocationMatch>

WSGIDaemonProcess keystone-admin processes=5 threads=1 user=www-
data display-name=%{GROUP}
WSGIProcessGroup keystone-admin
WSGIScriptAlias / /var/www/cgi-bin/keystone/admin
WSGIApplicationGroup %{GLOBAL}
ErrorLog /var/log/apache2/keystone.log
CustomLog /var/log/apache2/access.log combined
</VirtualHost>

# Workaround for missing path on RHEL6, see
# https://bugzilla.redhat.com/show_bug.cgi?id=1121019
WSGISocketPrefix /var/run/apache2

```

The **LocationMatch** tags represent a regular expression match on the URL, so that any URL containing `/Shib.../` will cause the enclosed `AuthType` (in this case `Shibboleth`) to be activated.

Note. `saml2` in the URL means that federated authentication should be via the SAMLv2 protocol, and `Shib.*` means that the IdP name should contain `Shib` in it. However, the current version of Keystone (Icehouse) does not check either of these fields, so they can effectively be set to anything. We will use this feature later when configuring Apache to use the Moonshot ABFAB plugin.

The path `/var/www/cgi-bin/keystone` should be configured according to Keystone's script location.

When the file above is created, enable it with the following command.

```
$> sudo a2ensite keystone.conf
```

Finally, restart the Apache2 service.

```
$> sudo service apache2 restart
```

When `apache2` is restarted, you should be able to access the URLs:

```
https://server.kent.ac.uk:5000/v2.0
```

```
https://server.kent.ac.uk:35357/v2.0
```

Please Note

Additional documentation about installing Keystone on Apache can be found in:
<http://docs.openstack.org/developer/keystone/apache-httpd.html>

2.2.7 Configuring mod_shib

In order to configure mod_shib, it's necessary to setup the IdP and SP information, such as endpoints and PKI trusted keys. Configuration files are stored in /etc/shibboleth. The files **shibboleth2.xml** and **metadata.xml** are important ones.

Please Note

For more information on mod_shib or mod_mellon configuration, refer to the following sites:

- http://docs.openstack.org/developer/keystone/configure_federation.html
- <http://docs.openstack.org/developer/keystone/extensions/shibboleth.html>

2.2.8 Populate Database – Part 2

In order to create users, tenants, roles, services and endpoints, first set up environment variables with endpoint and admin credentials. Replace the indicated values in blue with the ones used in your installation.

```
$> export OS_SERVICE_TOKEN=password
$> export OS_SERVICE_ENDPOINT=http://localhost:35357/v2.0
```

Create admin, demo, and service tenants.

```
$> keystone tenant-create --name admin --description "Admin Tenant"
$> keystone tenant-create --name demo --description "Demo Tenant"
$> keystone tenant-create --name service --description "Service
      Tenant"
```

Create admin and demo users.

```
$ keystone user-create --name admin --pass password --email
      admin@kent.ac.uk --tenant admin
$ keystone user-create --name demo --pass password --email
      demo@kent.ac.uk --tenant demo
```

It's very important to set the default tenant using "--tenant" parameter when creating users.

Create admin, _member_ and service roles.

```
$> keystone role-create --name admin
$> keystone role-create --name service
```

Assign admin role for admin user on tenants admin and demo.

```
$> keystone user-role-add --tenant admin --user admin --role admin
$> keystone user-role-add --tenant demo --user admin --role admin
```

Create Keystone service.

```
$> keystone service-create --name keystone --type identity
      --description "Openstack Identity"
```

Create Keystone endpoints.

```
$ keystone endpoint-create
```

```
--service-id
$(keystone service-list | awk '/ identity / {print $2}')
--publicurl http://service.kent.ac.uk:5000/v2.0
--internalurl http://service.kent.ac.uk:5000/v2.0
--adminurl http://service.kent.ac.uk:35357/v2.0
--region RegionOne
```

Please Note

Additional documentation about configuring a Keystone users, tenants, projects, services and endpoints can be found in the following sites.

<http://docs.openstack.org/juno/install-guide/install/apt/content/keystone-users.html>
<http://docs.openstack.org/juno/install-guide/install/apt/content/keystone-services.html>

2.3 Horizon

2.3.1 Downloading and installing the software

The Horizon software can be obtained from the following GIT repository.

https://github.com/ioram7/horizon/tree/federated_vo_management

Download it to `OPENSTACK_HOME` using the following command. Add “sudo” before the following commands when necessary.

```
$> cd OPENSTACK_HOME
$> git clone -b federated_vo_management
https://github.com/ioram7/horizon.git
```

To install it you should run the following commands:

```
$> cd horizon
$> sudo pip install -r requirements.txt
```

Create the settings file from the example.

```
$> cp openstack_dashboard/local/local_settings.py.example
openstack_dashboard/local/local_settings.py
```

Edit `local_settings.py` file and add the following lines:

```
OPENSTACK_HOST="service.kent.ac.uk"

OPENSTACK_KEYSTONE_URL="http://service.kent.ac.uk:5000/v2.0"

OPENSTACK_KEYSTONE_FEDERATED_SUPPORT = True
OPENSTACK_KEYSTONE_FEDERATED_URL =
    'http://service.kent.ac.uk:5000/v3'
OPENSTACK_DISCOVERY_USER = 'admin'
OPENSTACK_DISCOVERY_PASSWORD = 'password'
```

Add (or uncomment) the following lines:

```
OPENSTACK_API_VERSIONS = {
    "identity": 3,
}
```

Remove (or comment) the following lines:

```
OPENSTACK_HOST = "127.0.0.1"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
```

In order to Horizon enforce policies, add (or uncomment) the following lines:

```
POLICY_FILES_PATH = os.path.join("/etc", "keystone")
POLICY_FILES = {
    'identity': 'keystone_policy.json',
    'compute': 'nova_policy.json',
    'volume': 'cinder_policy.json',
    'image': 'glance_policy.json',
    'orchestration': 'heat_policy.json',
    #'network': 'neutron_policy.json',
}
```

Horizon needs that all policies are stored in a unique directory, the POLICY_FILES_PATH. You can create symbolic links to the original files like the example below:

```
$> cd /etc/keystone
$> ln -s /etc/keystone/policy.json keystone_policy.json
$> ln -s /etc/nova/policy.json nova_policy.json
$> ln -s /etc/cinder/policy.json cinder_policy.json
...
```

If the service is not installed, you can keep the line commented (as the neutron service in the example above).

Then run the following command to proceed with installation.

```
$> sudo python setup.py install
```

2.3.2 Configuring Apache2

Create a file **horizon.conf** into **/etc/apache2/sites-available/** directory using the template from the link below.

<http://git.openstack.org/cgit/openstack-dev/devstack/tree/files/apache-horizon.template>

Modify this file replacing the “tokens” according to the instructions and examples.

```
%USER% with www-data (Apache2's user)
%GROUP% with www-data (Apache2's group)
%HORIZON_DIR% with OPENSTACK_HOME/horizon
%APACHE_NAME% with apache2
```

Enable the site running the following command:

```
$> sudo a2ensite horizon.conf
```

Finally, let **apache2** user (**www-data**) be the owner of Horizon's files, create **DocumentRoot** directory for **horizon** and disable the default Apache2 site.

```
$> mkdir OPENSTACK_HOME/horizon/.blackhole
$> sudo chown -R www-data.www-data OPENSTACK_HOME/horizon
```

Execute the command below, if the file **/etc/apache2/sites-enabled/000-default** exists.

```
$> sudo a2dissite 000-default
```

Please Note

1. Additional documentation about installing Horizon on Apache can be found in: <http://docs.openstack.org/developer/horizon/topics/install.html>
2. Since Horizon needs modified python-keystoneclient and django-openstack-auth to run, it's recommended restarting apache only after these packages are installed .

2.4 Django_openstack_auth

The Django Openstack Auth library can be obtained from the following GIT repository.

https://github.com/ioram7/django_openstack_auth/tree/federated

It can be downloaded using the following command. We will use `OPENSTACK_HOME` to refer to the directory of your cloned repositories.

```
$> cd OPENSTACK_HOME
$> git clone -b federated
      https://github.com/ioram7/django_openstack_auth.git
```

To install it you should run the following commands:

```
$> cd django-openstack-auth
$> sudo pip install -r requirements.txt
```

Please Note

Horizon should have downloaded a new version of Openstack Auth as a dependency. It's important to make sure that the modified one is running.

The following line removes the old version and should be run before installing it.

```
$ sudo rm -rf /usr/local/lib/python2.7/dist-packages/django-
openstack_auth*
$ sudo rm -rf /usr/local/lib/python2.7/dist-packages/openstack_auth
```

```
$> sudo python setup.py install
$> sudo chown -R www-data.www-data
      OPENSTACK_HOME/django_openstack_auth
```

2.5 Keystoneclient

The Keystone Client library can be obtained from the following GIT repository.

https://github.com/ioram7/python-keystoneclient/tree/virtual_organisations

It can be downloaded using the following command. We will use `OPENSTACK_HOME` to refer to the directory of your cloned repositories.

```
$> cd OPENSTACK_HOME
$> git clone -b federated_vo_management
      https://github.com/ioram7/python-keystoneclient.git
```

Before installing it, you will need to delete old keystoneclient versions. For doing that, execute:

```
$> sudo rm -fr /usr/local/lib/python2.7/dist-packages/keystoneclient/
```

```
$> sudo rm -fr /usr/local/lib/python2.7/dist-  
packages/python_keystoneclient-1.0.0.dist-info/
```

To install it you should run the following commands:

```
$> cd python-keystoneclient  
$> sudo pip install -r requirements.txt
```

Please Note

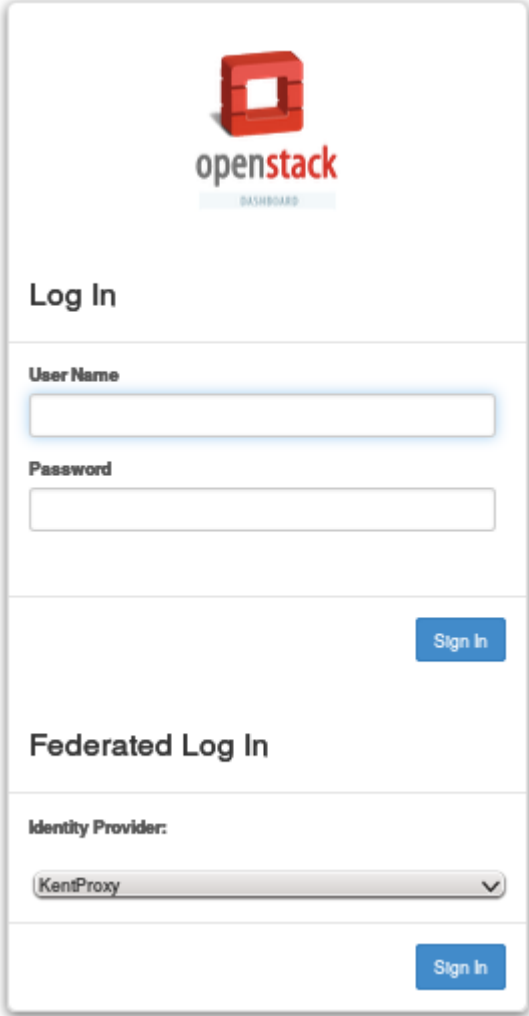
Keystone and Horizon should have downloaded a new version of Keystone Client as a dependency. It's important to make sure that the modified one is running. The following line removes the old version and should be run before installing it.

```
$ sudo rm -rf /usr/local/lib/python2.7/dist-packages/python_keystoneclient*  
$ sudo rm -rf /usr/local/lib/python2.7/dist-packages/keystoneclient
```

```
$> sudo python setup.py install
```

2.6 Configuring Federation

At this point, restart apache2 service and try to access horizon URL (eg.: <http://service.kent.ac.uk>). You should see a screen like this:



Log in using the defined admin user credentials.

Create Identity Providers by following the instructions.

On the left panel, choose “Identity → Identity Providers” and click on “+ Create Identity Provider” button.

Fill the form choosing an Id and a Description (any text) to the new IdP entry. It's important that the provider's Id matches with the “LocationMatch” entries of `/etc/apache2/sites-available/keystone.conf`. In our example, we should use an id that matches with `Shib.*`, for instance, “**Shibboleth_IdP1**”.

The **LocationMatch** tag is important if you want to use many modules in your configuration, for instance, `mod_moonshot` for `Abfab`.

Still on the left panel, choose “Groups” and then click on “+ Create Group” button.

Create a Group `ShibIdP1Users` with any description. A “Group Id” will be created for this group. Remember it, and create a mapping using it following the instructions below.

On the left panel, choose “Mappings” and then click on “+ Create Mapping” button.

Create any Id (eg.: **ShibMap**). The system will suggest to create an empty rule:

```
[{"local": [{"user": {"name": "{0}"}, {"group": {"id": "df444bdf3b69423db4c283fc280b58a3"}}, {"remote": {"type": "REMOTE_USER"}}]}
```

Each user that logs in using Shibboleth IdP will be assigned to this group.

Now, map the Identity Provider to the Mapping rule by adding a protocol. On the left panel, choose “Identity → Identity Providers” and on the “**Shibboleth_IdP1**” row, select “Manage Protocols” from the Actions list.

Click on the “+ Add Protocol” button and fill the form with the protocol id used in the LocationMatch URL (eg. **saml2**) and then select the mapping entry (**ShibMap**) from the list.

Finally, test if your IdP is working.

Click on the “Log Off” link (right-upper side) and you'll see the “Shibboleth_IdP1” Identity Provider listed on the “Federated Login” section. Select it and click on “Login” to be redirected to the IdP.

3 Configuration of Openstack with Abfab IdPs

This section describes how to install and configure Moonshot module to Apache. This module must be installed in the Openstack Server in order to add support to Abfab protocol.

It is assumed that Openstack (Keystone+Horizon) is already installed according to the instructions on section 0.

3.1 Requirements

It is assumed that Openstack (Keystone+Horizon) is already installed. Make sure that you updated Keystone to the latest version. Some files were included and some modified in order to add Abfab support.

Mod_shib needs to be installed in the Openstack service. Shib2 module for Apache MUST be DISABLED. Shibd service must be RUNNING.

This section presents the installation instructions for Moonshot module for Apache and to Configure a Moonshot IdP.

Despite there are instructions to make it work on Ubuntu 14.04, we were not successful on installing it. Therefore, **Ubuntu 12.04** and **Apache 2.2** are the recommended versions.

3.1.1 Django Openstack Auth – changes in the code

There are some hardcoded configuration in the following code that need to be changed in order to select the abfab protocol.

Edit the file:

[OPENSTACK_HOME](#)/django_openstack_auth/openstack_auth/views.py

Comment the line that refers to saml2 protocol, and add the ones referring to the abfab one just beneath it, according to the example below.

```
# identity = {'methods':['saml2'], 'saml2':{'id':  
request.POST.get('token')}}  
identity = {'methods':['abfab'], 'abfab':{'id':  
request.POST.get('token')}}
```

After these changes, go to the [OPENSTACK_HOME](#)/django_openstack_auth directory and run the command below to update the files.

```
S python setup.py install
```

Note 1. It's necessary to reinstall the python-keystoneclient libraries after installing the django_openstack_auth since it requires a newer version (1.0.0). Remember to remove them from the Python Library path (/usr/local/lib/python2.7/dist-packages).

Note 2. Remember to restart apache2 service after all libraries are correct.

3.1.2 Install Moonshot libraries on the Openstack Server

The Moonshot module for Apache can be downloaded and installed according to the instructions in the following web sites Besides the module itself, which must be

downloaded from the indicated git repository and compiled, it's also necessary to install some libraries.

Install libraries (steps 1 and 2):

<https://wiki.moonshot.ja.net/display/Moonshot/Install+Moonshot+Libraries+on+Ubuntu+12.04+LTS> (instructions for Ubuntu 12.04)

OR

<https://wiki.moonshot.ja.net/display/Moonshot/Install+Moonshot+Libraries+on+Ubuntu+14.04+LTS> (instructions for Ubuntu 14.04 – didn't work for us!)

Install download and install Moonshot module for Apache (steps 1 and 2):

<https://wiki.moonshot.ja.net/display/Moonshot/Apache+HTTPD+on+Debian+7>
(instructions for Debian7, also work with Ubuntu)

Please Note

Moonshot's module is not compatible with other federation modules, such as mod_shib.

The following line disables mod_shib plugin.

```
$ sudo a2dismod shib2
$ sudo service apache2 restart
```

3.1.3 Install and Configure a Moonshot IdP

An IdP running Abfab protocol is also required.

The following links have instructions for the Moonshot IdP installation and configuration. They were tested on a Debian 7 server.

<https://wiki.moonshot.ja.net/display/Moonshot/Install+an+IdP+on+Debian+7>

<https://wiki.moonshot.ja.net/display/Moonshot/Configure+an+Identity+Provider>

In the Moonshot IdP Server, the following code adds a SAML assertion in the FreeRadius authentication reply and should be inserted in the configuration file `/etc/freeradius/sites-enabled/abfab-tr-idp`, inside and at the end of the post-auth section.

```
if (request:GSS-Acceptor-Service-Name != 'trustidentity') {
    if ( (&reply:Moonshot-Realm-TargetedId) && !(&reply:User-Name) ) {

        update reply {
            User-Name := "%{reply:Moonshot-Realm-TargetedId}"
            # Erase the incoming SAML assertion
            SAML-AAA-Assertion !* 0x00
        }
    }
    if ( (&reply:User-Name) && !(&reply:SAML-AAA-Assertion[*]) )
    {
        update reply {
            SAML-AAA-Assertion = "<saml:Assertion
xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
IssueInstant='2011-03-19T08:30:00Z' ID='moonshot' Version='2.0'>"
            SAML-AAA-Assertion += "<saml:Conditions
NotOnOrAfter='2015-03-19T08:30:00Z' />"
        }
    }
}
```

```

        SAML-AAA-Assertion +=
"<saml:Issuer>urn:mace:incommon:osu.edu</saml:Issuer>"
        SAML-AAA-Assertion += "<saml:Subject><saml:NameID
Format='urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent'>{%reply:User-Name}</saml:NameID></saml:Subject>"
        SAML-AAA-Assertion += "<saml:AttributeStatement>"
        SAML-AAA-Assertion += "<saml:Attribute
NameFormat='urn:oasis:names:tc:SAML:2.0:attrname-format:uri'
Name='studentcard'>"
        SAML-AAA-Assertion +=
"<saml:AttributeValue>Student</saml:AttributeValue></saml:Attribute>"
        SAML-AAA-Assertion += "<saml:Attribute
NameFormat='urn:oasis:names:tc:SAML:2.0:attrname-format:uri'
Name='uid'>"
        SAML-AAA-Assertion +=
"<saml:AttributeValue>{%reply:User-
Name}</saml:AttributeValue></saml:Attribute>"
        SAML-AAA-Assertion += "<saml:Attribute
NameFormat='urn:oasis:names:tc:SAML:2.0:attrname-format:uri'
Name='affiliation'><saml:AttributeValue>kent</saml:AttributeValue></s
aml:Attribute>"
        SAML-AAA-Assertion += "</saml:AttributeStatement>"
        SAML-AAA-Assertion += "</saml:Assertion>"
    }
}

```

3.2 Configuration of the Moonshot Plugin

This section describes how to configure Apache's Moonshot Plugin in the Openstack Server.

Create the file **/etc/radsec.conf** according to the example:

```

realm gss-eap {
    type = "TLS"
    cacertfile = "/etc/freeradius/certs/ca.pem"
    certfile = "/etc/freeradius/certs/client.pem"
    certkeyfile = "/etc/freeradius/certs/client.key"
    disable_hostname_check = yes
    server {
        hostname = "moonshot.sec.cs.kent.ac.uk"
        service = "2083"
        secret = "radsec"
    }
}

```

The **cacertfile** attribute must contain the radius server CA certificate, **certfile** must contain the Openstack Server's certificate signed by the CA, and **certkeyfile** must contain the Openstack Server's private key. These files are used for the TLS communication with the Moonshot server, whose address is represented by the **server→hostname** attribute.

3.3 Configuration in Apache for the Moonshot Plugin

Backup your Keystone's site configuration file (**keystone.conf**) into **/etc/apache2/sites-available**, and create a new one according to the example below.

```

Listen 5000
Listen 35357

# Standard Keystone endpoint
<VirtualHost *:5000>

    # Protect the Keystone federated endpoint for abfab
    <LocationMatch /v3/OS-
FEDERATION/identity_providers/Abfab.*?/protocols/abfab/auth>
        AuthType Negotiate
        Require valid-user
    </LocationMatch>
    WSGIDaemonProcess keystone-public processes=5 threads=1 user=
www-data display-name=%{GROUP}
    WSGIProcessGroup keystone-public
    WSGIScriptAlias / /var/www/cgi-bin/keystone/main
    WSGIApplicationGroup %{GLOBAL}
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>

# Admin Keystone endpoint
<VirtualHost *:35357>

    # Protect the Keystone federated endpoint for abfab
    <LocationMatch /v3/OS-
FEDERATION/identity_providers/Abfab.*?/protocols/abfab/auth>
        AuthType Negotiate
        Require valid-user
    </LocationMatch>
    WSGIDaemonProcess keystone-admin processes=5 threads=1 user=www-
data display-name=%{GROUP}
    WSGIProcessGroup keystone-admin
    WSGIScriptAlias / /var/www/cgi-bin/keystone/admin
    WSGIApplicationGroup %{GLOBAL}
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>

# Workaround for missing path on RHEL6, see
# https://bugzilla.redhat.com/show_bug.cgi?id=1121019
WSGISocketPrefix /var/run/apache2

```

This configuration tells Apache that the ABFAB Moonshot protocol (AuthType Negotiate) will be used to protect the Keystone endpoint for the Identity Provider starting with “**Abfab**” and using abfab protocol.

3.4 Configuring Abfab in Keystone

Edit the file “**/etc/keystone/keystone.conf**” and modify the following lines in the **[auth]** section.

```

methods=external,password,token,saml2,abfab
saml2=keystone.auth.plugins.mapped.Mapped
abfab=keystone.auth.plugins.mapped.Mapped

```

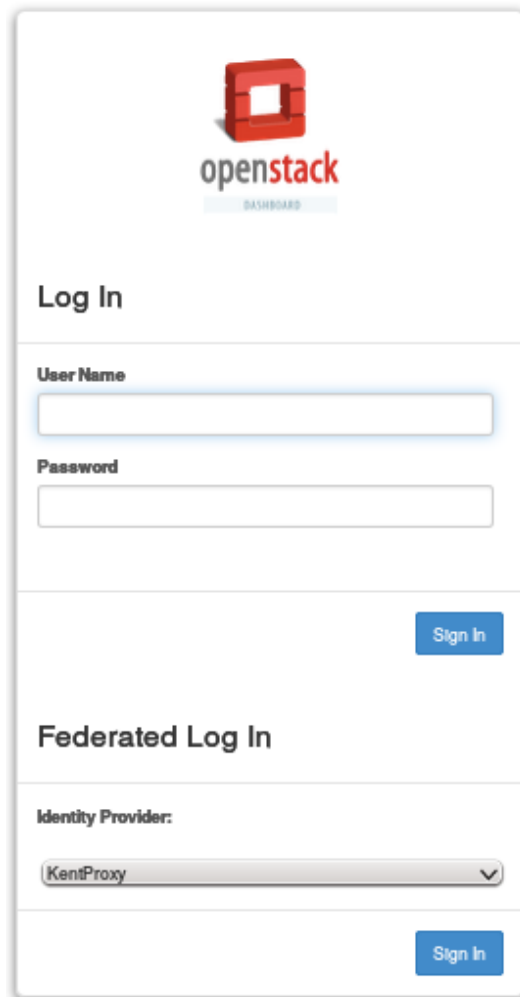
After the above configuration, restart Apache:

```

sudo service apache2 restart

```

At this point, restart apache2 service and try to access horizon URL (eg.: <http://service.kent.ac.uk>). You should see a screen like this:



The screenshot shows the OpenStack Dashboard login interface. At the top is the OpenStack logo with the word 'openstack' in red and 'DASHBOARD' in a light blue box below it. Below the logo is the heading 'Log In'. Underneath are two input fields: 'User Name' and 'Password'. To the right of the 'Password' field is a blue 'Sign In' button. Below the 'Log In' section is the 'Federated Log In' section. It features a label 'Identity Provider:' followed by a dropdown menu currently showing 'KentProxy'. To the right of the dropdown is another blue 'Sign In' button.

Log in using the admin user credentials.

Create the Abfab Identity Provider by following the instructions.

On the left panel, choose “Identity → Identity Providers” and click on “+ Create Identity Provider” button.

Fill the form choosing an Id and a Description (any text) to the new IdP entry. It's important that the provider's Id matches with the “LocationMatch” entries of `/etc/apache2/sites-available/keystone.conf`. In our example, we should use an id that matches with **Abfab.***, for instance, “**Abfab_JaNet**”.

Still on the left panel, choose “Groups” and then click on “+ Create Group” button.

Create a Group **JaNetUsers** with any description. A “Group Id” will be created for this group. Remember it, and create a mapping using it following the instructions below.

On the left panel, choose “Mappings” and then click on “+ Create Mapping” button.

Create any Id (eg.: **JaNetMap**). The system will suggest to create an empty rule:
`[{"local": [{"user": {"name": "{0}"}, {"group": {"id": "df444bdf3b69423db4c283fc280b58a3"}}], "remote": [{"type": "REMOTE_USER"}]}`

Each user that logs in using the JaNet IdP will be assigned to this group.

Now, map the Identity Provider to the Mapping rule by adding a protocol. On the left panel, choose “Identity → Identity Providers” and on the “**Abfab_JaNet**” row, select “Manage Protocols” from the Actions list.

Click on the “+ Add Protocol” button and fill the form with the protocol id used in the LocationMatch URL (eg. **abfab**) and then select the mapping entry (**JaNetMap**) from the list.

An additional step is necessary in order to assign a Project and a Role to the user group. This project will be presented to the federated user when it logs in the system.

First, choose “Projects → Create Project” from the left panel to create a new Project (eg. **JaNetProject**). Leave the Domain default information as suggested, and fill the other informations (name, description, and quota details are mandatory).

We currently cannot assign the group to the project using the Horizon UI (its probably a bug). To do this, we used a mysql command line interface (CLI). Access a terminal on the Openstack server and type:

```
$ mysql keystone -u root -p
Type root user's password.
```

```
mysql> insert into assignment select "GroupProject", g.id, p.id, r.id, 0 from `group` as
g, project as p, role as r where g.name = "JaNetUsers" and p.name =
"JaNetProject" and r.name = "_member_1";
```

This makes every member of the **JaNetUsers** group a member of the **JaNetProject**. Note that the *group* table is quoted with the grave accent symbol (`) since it is a reserved word in mysql.

Finally, test if your IdP is working.

Click on the “Log Off” link (right-upper side) and you’ll see the “Abfab_Janet” Identity Provider listed on the “Federated Login” section. Select it and click on “Login” to be redirected to the IdP.

3.5 Moonshot SSP

For testing Openstack with Abfab authentication, it’s required to use a Window’s client with Internet Explorer 11 and also to have Monshot’s SSP installed.

Monshot’s SSP is provided by **ja.net**². It is very easy to install by running the executable and following the Wizard instructions. It’s important to notice if you are installing the appropriate version for your architecture (32 or 64 bits).

¹ `_member_` is the default member role name on Openstack systems.

When everything is set up, open IEv11 and try to reach your Openstack server. Choose the “**Abfab_JaNet**” provider and you should be prompted for Username and Password. Type you username including your realm (<username>@<realm>) and password and you should be logged in. If it succeeds, you will be presented a page to choose a project.

4 Configuring OpenStack Privileges to a VO Role

The VO is represented as a Domain and a VO Role as a Group inside Keystone. The VO Role’s group is automatically created at the time the VO Role is created. The name of the Group is the same name as the VO Role. Similarly, a new domain is created when the first VO Role of a VO is created. The Domain also receives the same name as the VO.

Since Openstack uses a Role-Based Access Control (RBAC), an OpenStack Role needs to be assigned to the VO Role’s Group to give any OpenStack privileges to this VO Role. The VO Role also needs to be authorised to perform certain actions on resources.

Section 4.1 explains how to assign Openstack Roles to VO Roles and section 4.2 explains how to give privileges to an Openstack Role.

4.1 Assign Openstack Roles to VO Roles

In order to assign an OpenStack role to a VO Role/group in Keystone, execute the commands as in the example below. The OpenStack roles **admin** and **_member_** are associated to the VO Role/group **VO_Role_Name** on the project **JaNetProject** in the domain **VO_Name**.

Enter MySQL CLI:
\$ mysql keystone -u root -p
Type root user’s password.

Create the assignment VO_Role_Name – JaNetProject -> admin role:
mysql> insert into assignment select "GroupProject", g.id, p.id, r.id, 0 from `group` as g, project as p, role as r where g.name = "**VO_Role_Name**" and p.name = "**JaNetProject**" and r.name = "**admin**";

Create the assignment VO_Role_Name – JaNetProject -> member role:
mysql> insert into assignment select "GroupProject", g.id, p.id, r.id, 0 from `group` as g, project as p, role as r where g.name = "**VO_Role_Name**" and p.name = "**JaNetProject**" and r.name = "**_member_**";

Create the assignment VO_Role_Name – VO_Name domain -> admin role:
mysql> insert into assignment select "GroupDomain", g.id, d.id, r.id, 0 from `group` as g, domain as d, role as r where g.name = "**VO_Role_Name**" and d.name = "**VO_Name**" and r.name = "**admin**";

Create the assignment VO_Role_Name – VO_Name domain -> member role:

```
mysql> insert into assignment select "GroupDomain", g.id, d.id, r.id, 0 from `group`
as g, domain as d, role as r where g.name = "VO_Role_Name" and d.name =
"VO_Name" and r.name = "_member_";
```

Note that the *group* table are quoted with the grave symbol (`) since it is a reserved word in mysql.

Now that members of the VO Role are mapped to the Openstack roles when logged in, we need to give permissions to the role so that it can perform actions.

4.2 Assign Authorisation Policies to Openstack Roles

Authorisation policies in Openstack are defined in **policy.json** files. There is one policy file per service (eg. keystone, nova, glance, ...), which is usually stored in the directory **/etc/<service_name>**.

Horizon needs to access these files to present its graphic interface properly. So, remember to refer to each of these files in the Horizon configuration (*local_settings.py* file) according to the instructions detailed in section 2.3.1.

The example below explains how to give “administrative privileges on VOs” to a given Openstack role called “**VoAdmin**”. Let’s say that the VO Role “Admin” in the VO named “Classe” is assigned to this VO Role (following the instructions from section 4.1).

Note that “administrative privileges on VOs” means that these users can create VOs, and manage VOs. For instance, they will be allowed to approve requests from users to join a VO Role. In order to have administrative privileges on all Openstack services and operations, it would be sufficient to assign the Openstack role “admin” to the VO Role as in the example of the previous section. However this typically gives too many privileges to a VO administrator.

Edit the file **/etc/keystone/policy.json** and add/change lines according to the instructions below.

1. At the end of the first block, create a label “**vo_admin**” including the openstack role “**VoAdmin**” and also the Openstack admin. This is useful so that we don’t need to repeat the “or” condition on every policy rule.

```
"vo_admin": "role:VoAdmin or rule:admin_required",
```

2. In the middle of the file, change the lines below to give the “vo_admin” group to perform actions on Domains and Groups, which represents VO and VO Roles respectively.

```
"identity:get_domain": "rule:vo_admin",
"identity:list_domains": "rule:vo_admin",
"identity:create_domain": "rule:vo_admin",
"identity:update_domain": "rule:vo_admin",
"identity:delete_domain": "rule:vo_admin",
```

...

```
"identity:get_group": "rule:vo_admin",
"identity:list_groups": "rule:vo_admin",
```

```

"identity:list_groups_for_user": "rule:vo_admin or
                                rule:admin_or_owner",
"identity:create_group": "rule:vo_admin",
"identity:update_group": "rule:vo_admin",
"identity:delete_group": "rule:vo_admin",
"identity:list_users_in_group": "rule:vo_admin",
"identity:remove_user_from_group": "rule:vo_admin",
"identity:check_user_in_group": "rule:vo_admin",
"identity:add_user_to_group": "rule:vo_admin",

```

3. Also edit this line according to the example. It is necessary to retrieve the user name and present them in the UI.

```

"identity:get_user": "rule:vo_admin",

```

4. Finally, at the end of the file, add or modify the following entries. Don't forget to add a comma at the end of the original last line!

```

.....,
"identity:vo_admin": "rule:vo_admin",

"identity:create_vo_role": "rule:vo_admin",
"identity:list_vo_roles": "rule:vo_admin",
"identity:get_vo_role": "rule:vo_admin",
"identity:delete_vo_role": "rule:vo_admin",
"identity:update_vo_role": "rule:vo_admin",

"identity:add_user_to_vo_role": "rule:vo_admin",
"identity:list_vo_roles_members": "rule:vo_admin",
"identity:get_vo_role_member": "rule:vo_admin",
"identity:remove_vo_role_membership_from_user": "rule:vo_admin",
"identity:switch_vo_role_for_user": "rule:vo_admin",

"identity:list_vo_requests": "rule:vo_admin",
"identity:decline_vo_request": "rule:vo_admin",
"identity:approve_vo_request": "rule:vo_admin",

"identity:get_vo_blacklist": "rule:vo_admin",
"identity:remove_user_from_blacklist": "rule:vo_admin",

"identity:join_vo_role": "",
"identity:list_my_vo_roles": "",
"identity:check_vo_membership_status": "",
"identity:resign_from_role": ""

```

After that, restart apache and log in as a member of the VO Role, and you will be able to access VO administration tasks in Horizon.

5 Usage



This section contains screen shots and instructions describing the usage of the Horizon Interface for new or modified components only. For usage instructions for standard operations, please see OpenStack documentation [here](#)

5.1 Federated Horizon

5.1.1 Logging in

The screenshot shows the OpenStack Dashboard login interface. At the top, there is the OpenStack logo and the word 'openstack' in a stylized font, with 'DASHBOARD' written below it. The main heading is 'Log In'. Below this heading are two input fields: 'User Name' and 'Password'. To the right of the 'Password' field is a blue 'Sign In' button. Below the 'Log In' section is the 'Federated Log In' section. It features a dropdown menu for 'Identity Provider' with 'KentProxy' selected. Below the dropdown menu is another blue 'Sign In' button.

To login using federation, select the desired Identity Provider and press “Sign In”, alternatively username and password can be used.

5.1.2 Identity Provider Management

5.1.2.1 View Identity Providers

You can view a list of Identity Providers. The following details are displayed:

- Identity Provider ID – the name or ID given to the identity provider (IdP) during creation.
- Description – An optional description for this IdP.
- Supported Protocols - a comma-separated list of supported protocols.
- Action drop down for managing this IdP. Includes Edit, Manage Protocols and Delete.

Identity Providers

Identity Providers				
Filter <input type="text"/>		Filter <input type="text"/>	<input type="button" value="+ Create Identity Provider"/> <input type="button" value="X Delete Identity Providers"/>	
<input type="checkbox"/>	Identity Provider ID	Description	Supported Protocols	Actions
<input type="checkbox"/>	KentProxy	Kent ISSRG group proxy server	saml2	Edit <input type="button" value="v"/>

Displaying 1 item

5.1.2.2 Add Identity Providers

Press the Create Identity Provider button to open the Add Identity Provider dialog:

Create Identity Provider ✕

ID *

Description:

From here you can create a new identity provider.

Description

✕

Enter the ID and description for the new IdP and then press Create Identity Provider.

5.1.2.3 Edit Identity Providers

Pressing the “Edit” action button next to an IdP allows you to modify the description of the IdP:

5.1.2.4 Delete Identity Providers

Pressing the “Delete” action will delete the IdP specified, alternatively, ticking the check box next to one or more IdPs and then pressing the “Delete Identity Providers” button will delete all selected IdPs.

5.1.2.5 Manage Protocols for an Identity Provider

Pressing the “Manage Protocols” action will open the protocol management interface:

Protocol Management: KentProxy

Protocol ID	Mapping ID	Actions
<input type="checkbox"/> saml2	kentmapping	Edit

Displaying 1 item

A list of protocols supported by this IdP will be displayed with the following information:

- Protocol ID – the identifier for this protocol.
- Mapping ID – the attribute mapping policy that should be used when users from this IdP authenticate using this protocol.
- Action drop down for managing this Protocol. Includes Edit, and Delete.

5.1.2.6 Add protocol

To Add a new protocol, cpress the Add Protocol button in the top right to open the add protocol dialog:

Add Protocol

Protocol *

Mapping ID *

Description:
From here you can add a protocol to an identity provider

Cancel Add Protocol

Enter the ID for the new Protocol and choose from the available mapping policies, then press Add Protocol.

5.1.2.7 Edit Protocol

Pressing the Edit action next to a protocol opens the edit dialog for protocols that allows the associated mapping policy to be changed:

Update Protocol

Protocol *

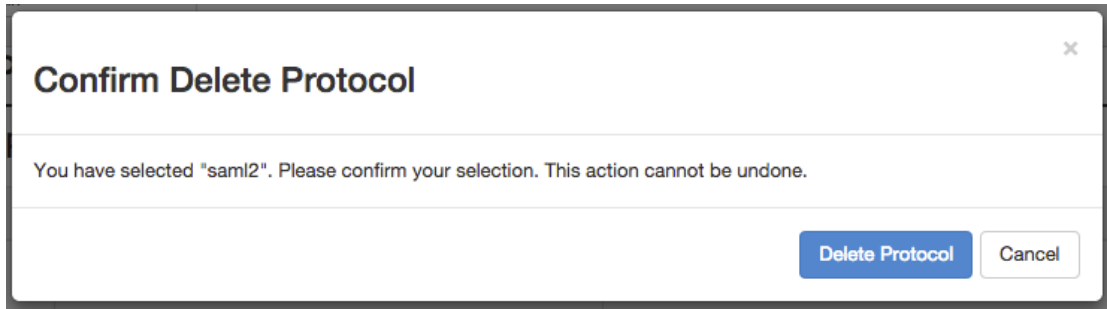
Mapping ID *

Description:
From here you can modify the mapping for a protocol on an identity provider

Cancel Update Protocol

5.1.2.8 Delete Protocol

Pressing the “Delete” action will delete the Protocol specified, alternatively, ticking the check box next to one or more Protocols and then pressing the “Delete Protocols” button will delete all selected Protocols:

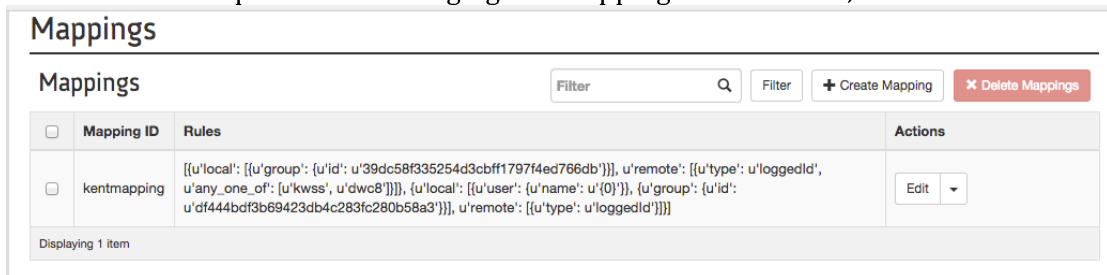


5.1.3 Mapping Management

5.1.3.1 View Mappings

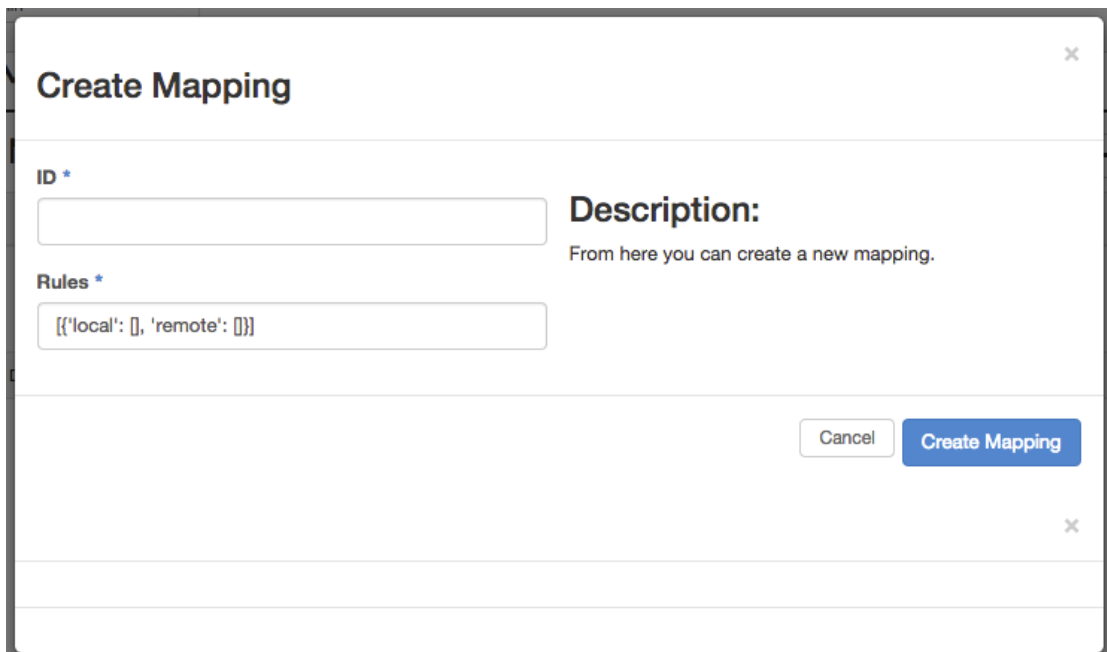
You can view a list of Mapping Policies. The following details are displayed:

- Mapping ID – the name or ID given to the Mapping during creation.
- Rules – The JSON mapping rules for this policy.
- Action drop down for managing this Mapping. Includes Edit, and Delete



5.1.3.2 Add Mapping

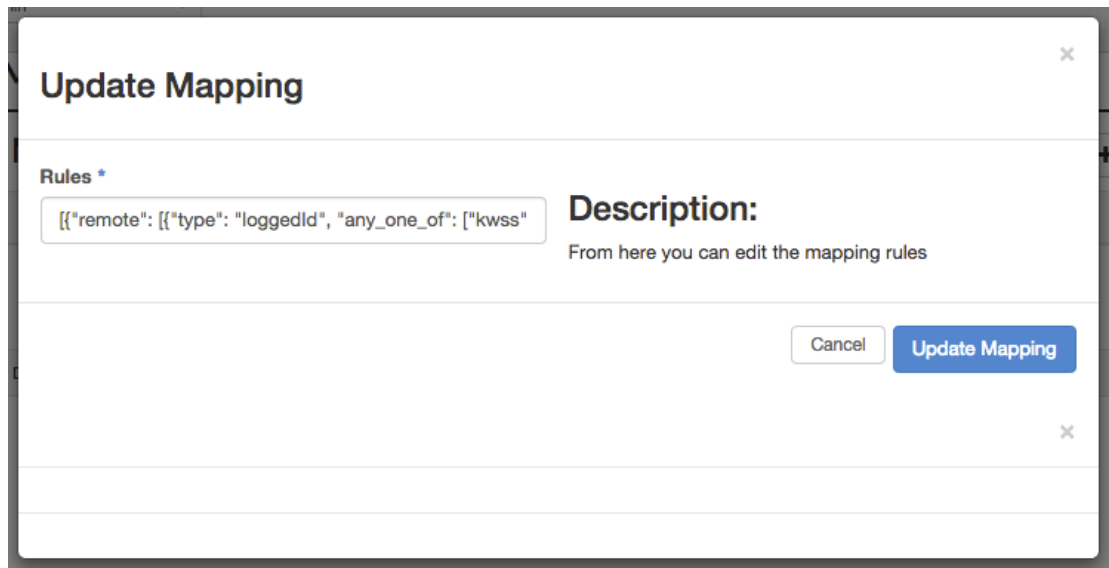
Pressing Create Mapping will open the create mapping dialog:



Enter the ID for this mapping and the JSON rules which should be used and then press Create Mapping.

5.1.3.3 Edit Mapping

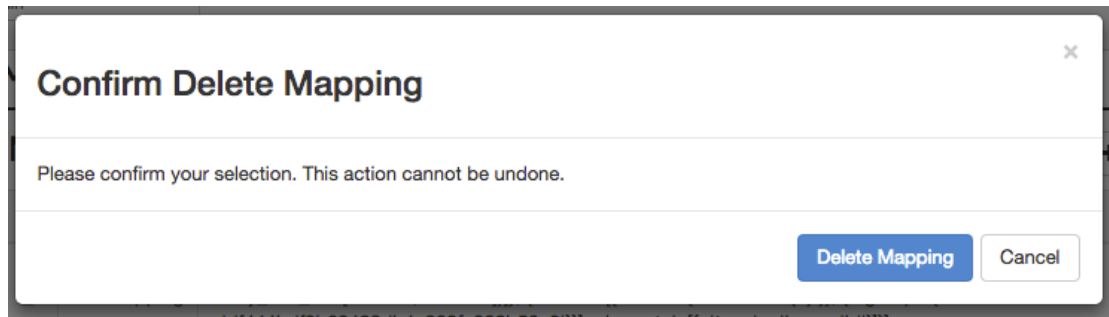
Pressing the Edit action next to a Mapping will open the Edit Mapping Dialog which allow the rules of the policy to be modified:



The screenshot shows a dialog box titled "Update Mapping" with a close button (X) in the top right corner. The dialog is divided into two main sections. The first section is labeled "Rules *" and contains a text input field with the following JSON rule: `{{"remote": [{"type": "loggedId", "any_one_of": ["kwss"]`. The second section is labeled "Description:" and contains the text "From here you can edit the mapping rules". At the bottom right of the dialog, there are two buttons: "Cancel" and "Update Mapping".

5.1.3.4 Delete Mapping

Pressing the "Delete" action will delete the Mapping specified, alternatively, ticking the check box next to one or more Mappings and then pressing the "Delete Mappings" button will delete all selected Mappings.



The screenshot shows a dialog box titled "Confirm Delete Mapping" with a close button (X) in the top right corner. The dialog contains a single line of text: "Please confirm your selection. This action cannot be undone." At the bottom right of the dialog, there are two buttons: "Delete Mapping" and "Cancel".

6 Useful Links

Description	Link
GIT documentation	http://git-scm.com/
Openstack documentation	http://docs.openstack.org/
Moonshot documentation	https://wiki.moonshot.ja.net/display/Moonshot/
