Federated Access to OpenStack

January 21st, 2016

GRUPPO RETI

Business & IT Consulting

Innovation & Efficiency

www.reti.it

Goal

- Federate the authentication and authorization of OpenStack:
  - permit access to users identified by Identity Providers in eduGAIN
  - obtain group information on users to assign their access rights on the projects

- Integrate OpenStack and standalone Attribute Authorities
  - to split the processes of authentication and of authorization
  - to leverage the process shift happening in many NRENs and identity federations

Acknowledgement

- This work has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 691567 (GN4-1).

- This work has been realized by:
  - Mihály Héder (MTA SZTAKI)
  - Szabolcs Tenczer (MTA SZTAKI)
  - Andrea Biancini (formerly GARR, now Reti)

- Article available here: *Héder Mihály, Szabolcs Tenczer, and Andrea Biancini. "Collaboration between SAML Federations and OpenStack Clouds." arXiv preprint arXiv:1510.04017 (2015).*

What's already there…

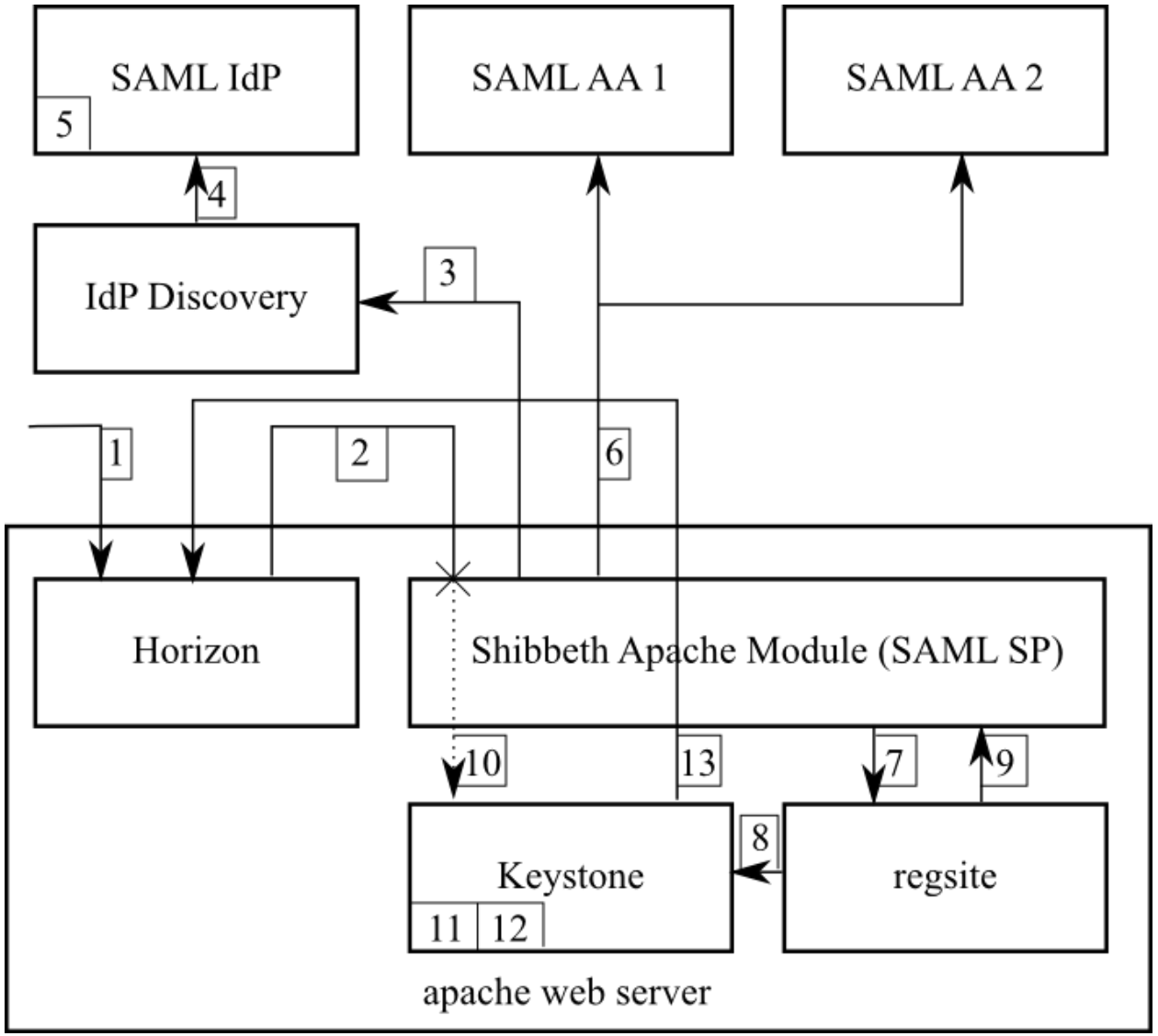
- Keystone has already the possibility to be configured as a SAML SP
  - David Chadwick from University Kent initiated a project to SAML-enable keystone and OpenStack back in 2012
  - In 2015 OpenStack extended this solution, with contributions from RedHat, CERN and IBM (websso)

The problem

- The OpenStack websso solution works in two different ways:
  - With ephemeral users, so users which has not been provisioned a-priori in keystone. This solution has a smaller footprint (no pre-provisioning needed) but can create some complication in authorization (yet ephemeral users inherit group privileges).
  - With users already provisioned into Keystone before the request for access to the horizon web interface. If the federation is not configured to support ephemerals, users which are not already present in Keystone, would successfully login into Shibboleth federation middleware, just to instantly get access denied with an error message from Keystone.

- We wanted to permit OpenStack websso solution to manage authorization information coming from attributes retrieved from the user SAML session.

Proposed architecture



- Keystone must be configured as an SP (so with HTTPs), we have chosen to have it configured inside Apache httpd.
- The SP is configured to retrieve information from an independent Attribute Authority in the Federation.
- A new component (called regsite) is installed to take care of user provisioning.

Engineering properties of the solution

- Encapsulation of new functionality within a self-contained module
  - very important to permit compatibility with new releases of OpenStack (which has a half-year release cycle)
- Reuse of mature components
  - to fully manage protocols and standards without having to fully re-implement
- Full compatibility with SAML
  - to leverage on existing technologies within Identity Federations
- Delegation of administration
  - user provisioning and authorization are delegated to externally managed systems

Main components code and configuration

- The authentication submodule keystone needs to be configured as a Service Provider to a federation. This can be done activating the “websso" module of keystone.
- A webpage needs to be created to manage user/project creation, this page will operate as a standalone "signup page" web app (working title: regsite). This registration page can be automatically invoked after user login with a specific "post login" hook provided by the Shibboleth SP.
- Horizon needs to authenticate users with a web-SSO profile, usually provided by Shibboleth SP.

Configuration of Keystone – Apache

```
<VirtualHost *:5000>
    SSLEngine on
    [...]

    WSGIScriptAlias / /var/www/cgi-bin/keystone/main
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On

    <Location /v3/auth/OS-FEDERATION/websso/saml2>
        authtype shibboleth
        shibRequestSetting requireSession 1
        require valid-user
    </Location>
</VirtualHost>
```

Configuration of Keystone – keystone.conf

```
[DEFAULT]
public_endpoint = https://openstack.localdomain:5000

[auth]
methods = external,password,token,oauth1,saml2
saml2 = keystone.auth.plugins.mapped.Mapped

[federation]
remote_id_attribute = 'Shib-Identity-Provider'
trusted_dashboard = https://openstack.localdomain/dashboard/auth/websso/
sso_callback_template = /etc/keystone/sso_callback_template.html

[ssl]
enable=True
```

Registration of IdP in keystone OS_FEDERATION


```
{
  "links": {"self": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/identity_providers", "previous": null, "next": null},
  "identity_providers": [{
    "remote_ids": ["https://testidp.localdomain/idp/shibboleth"],
    "enabled": true,
    "id": "RetiLab",
    "links": {
      "self": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/identity_providers/RetiLab",
      "protocols": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/identity_providers/RetiLab/protocols"
    },
    "description": "IdPs of the RetiLab federation"
  }]
}
```

Registration of Protocols in keystone OS_FEDERATION


```
{
  "protocol": {
    "mapping_id": "shib",
    "id": "saml2",
    "links": {
      "self": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/identity_providers/RetiLab/protocols/saml2",
      "identity_provider": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/identity_providers/RetiLab"
    }
  }
}
```

Registration of Mapping in keystone OS_FEDERATION

```
{
  "mapping": {
    "rules": "[{\"remote\": [{\"type\": \"eppn\"}], \"local\": [{\"user\":
{\"domain\": {\"id\": \"default\"}, \"type\": \"local\", \"name\":
\"{0}\"}}]}]",
    "id": "shib",
    "links": {
      "self": "https://openstack.localdomain:5000/v3/OS-
FEDERATION/mappings/shib"
    }
  }
}
```

Installation of the regsite web application

- Regsite is a web application implemented in python.
- It can be configured on the same Apache virtualhost of keystone.

```
WSGIPassAuthorization Off
WSGIScriptAlias /regsite /opt/openstack_regsite/wsgi.py

<Directory /opt/openstack_regsite>
    Options all
    AllowOverride all
    Require all granted
</Directory>
```

Configuration of regsite

```
OPENSTACK_NAME = 'Test OpenStack for SAML integration'
OPENSTACK_URL = 'https://openstack.localdomain'
OPENSTACK_KEYSTONE_ADMIN_URL = 'http://openstack.localdomain:35357/v3'
OPENSTACK_KEYSTONE_ADMIN_TOKEN = 'd1410ad0ab162c016171'
SHIBBOLETH_NAME_ATTRIBUTE = 'eppn'
SHIBBOLETH_ENTITLEMENT_ATTRIBUTE = 'isMemberOf'
SHIBBOLETH_EMAIL_ATTRIBUTE = 'mail'
DEFAULT_DOMAIN_NAME = 'Default'
USER_ACCEPT_CREATION = True
```

What does regsite do?

- Regsite receives a federate identifier, an entitlement (authoritative information) and optionally a mail attribute from the Shibboleth middleware.
- From the entitlement information projects and roles are derived as follows:
  - the format of the entitlement is: *entilement_prefix:project:role*
  - divided by the colon (:), the segments are used to represent project and role that are resources in OpenStack
  - there might be other colons in the entitlement prefix, the software always uses the last two segments
  - in the case there are multiple entitlement attributes, all of them are used
- Regsite updated the projects and user in keystone with these information (creates new ones, update existing).

Configuration of the Shibboleth SP

```
<ApplicationDefaults entityID="https://openstack.localdomain/shibboleth"
                     REMOTE_USER="eppn"
                     sessionHook="/regsite"
                     signing="true">
[...]

<AttributeResolver type="SimpleAggregation" attributeId="eppn"
                   format="urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified">
    <Entity>https://grouper.localdomain/idp/shibboleth</Entity>
</AttributeResolver>

[...]
</ApplicationDefaults>
```

Horizon configuration

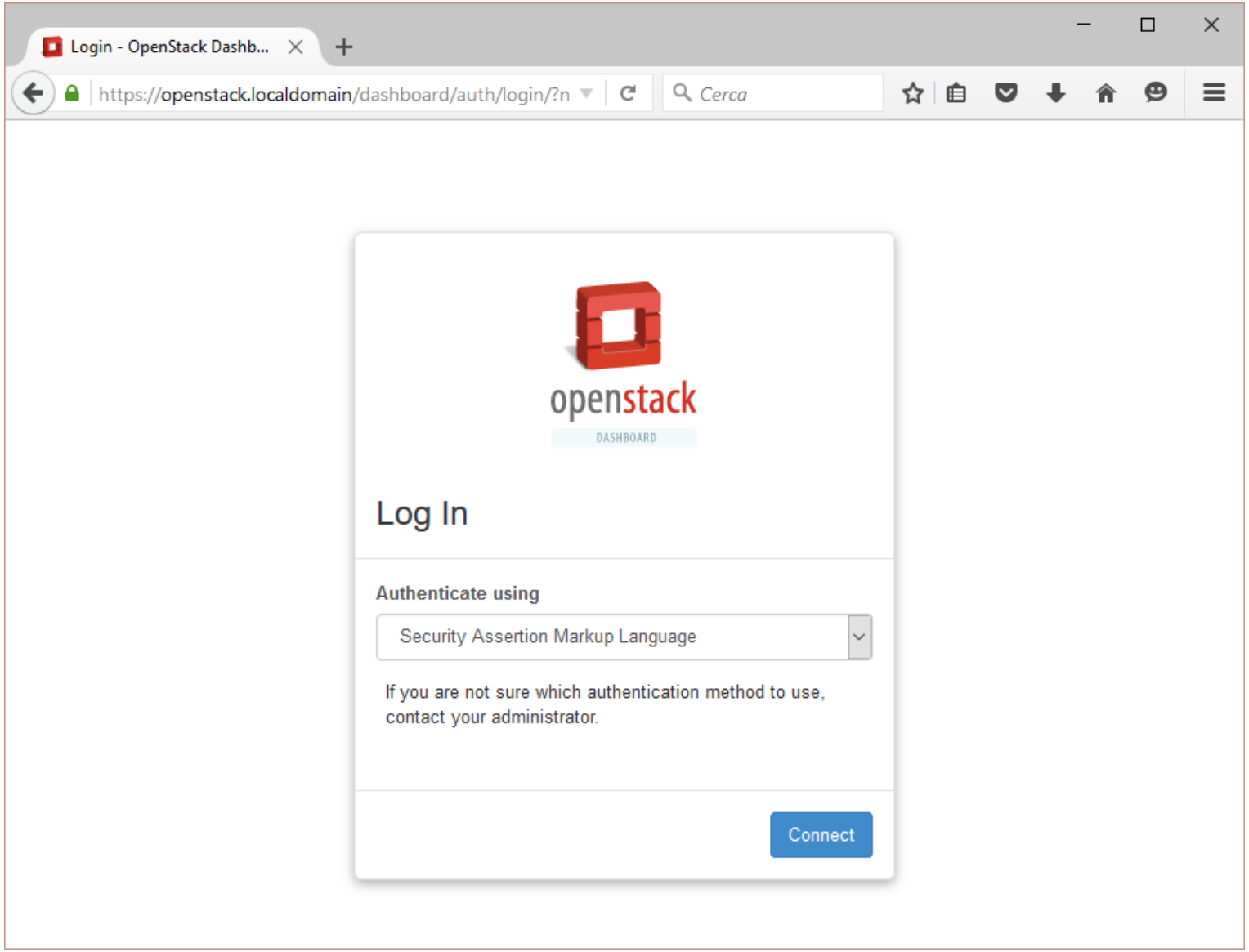
```
OPENSTACK_KEYSTONE_URL = "https://openstack.localdomain:5000/v3"

OPENSTACK_API_VERSIONS = {
    "identity": 3,
}

WEBSSO_ENABLED = True

WEBSSO_INITIAL_CHOICE = "saml2"

WEBSSO_CHOICES = (
    ("credentials", _("Keystone Credentials")),
    ("saml2", _("Security Assertion Markup Language")))
```

Example of a user access

- We will then try a login to se the whole process:
  - the user will try to access horizon web-interface (he will be able to choose between local access or SAML access) [1]
  - if the user will choose SAML, he will be redirected to the DS or the IdP for federated authentication
  - after login, in a transparent way, the SP will contact the AA to retrieve group membership for the user
  - if the authorization requires some modification to the keystone database to grant the correct access to the user, the OpenStack Registration Site will be presented [2]
  - the user can now choose to continue and, in this case, he will be granted access to horizon with the right authorization to the right projects [3]

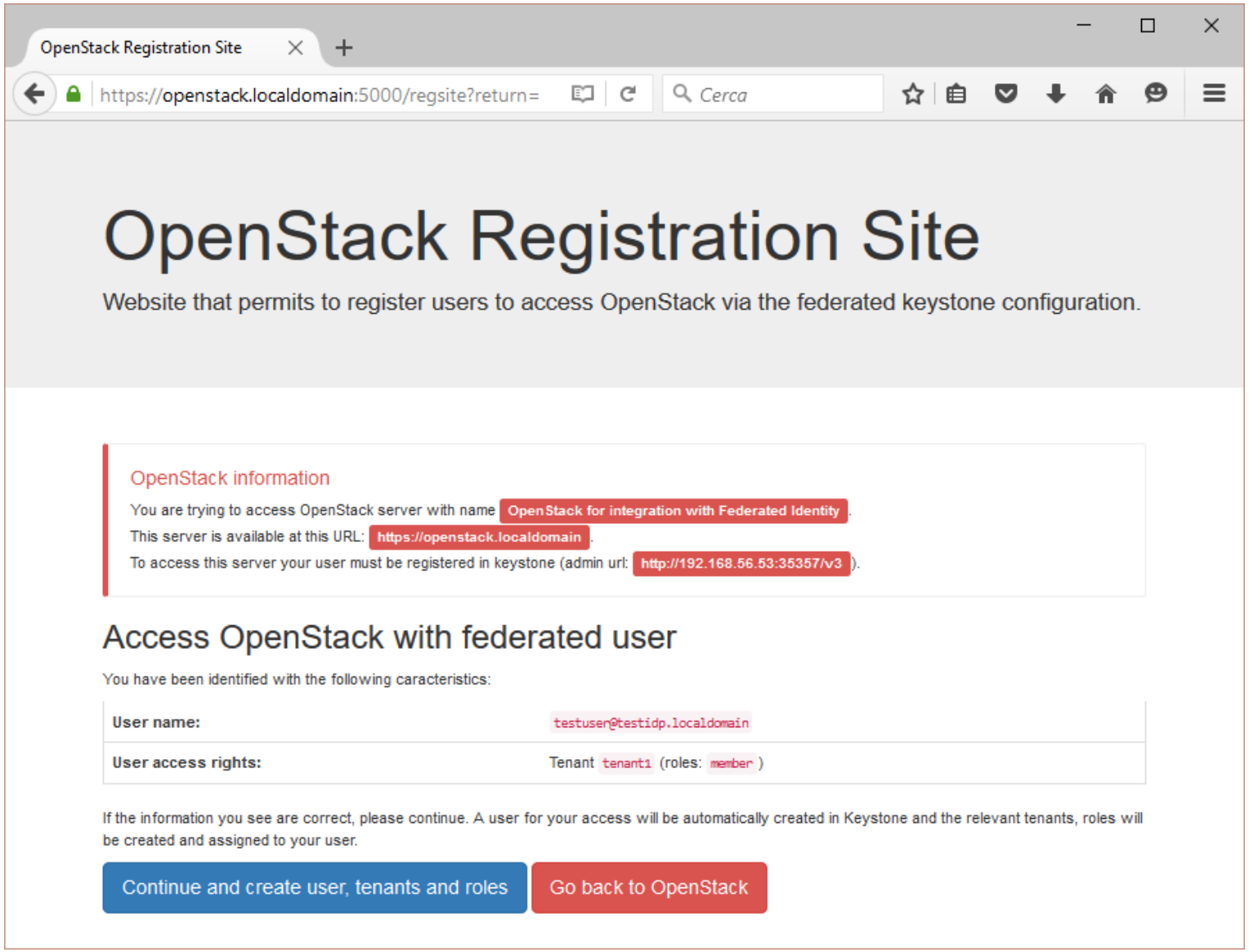
Live Demo

[1] Login to OpenStack horizon



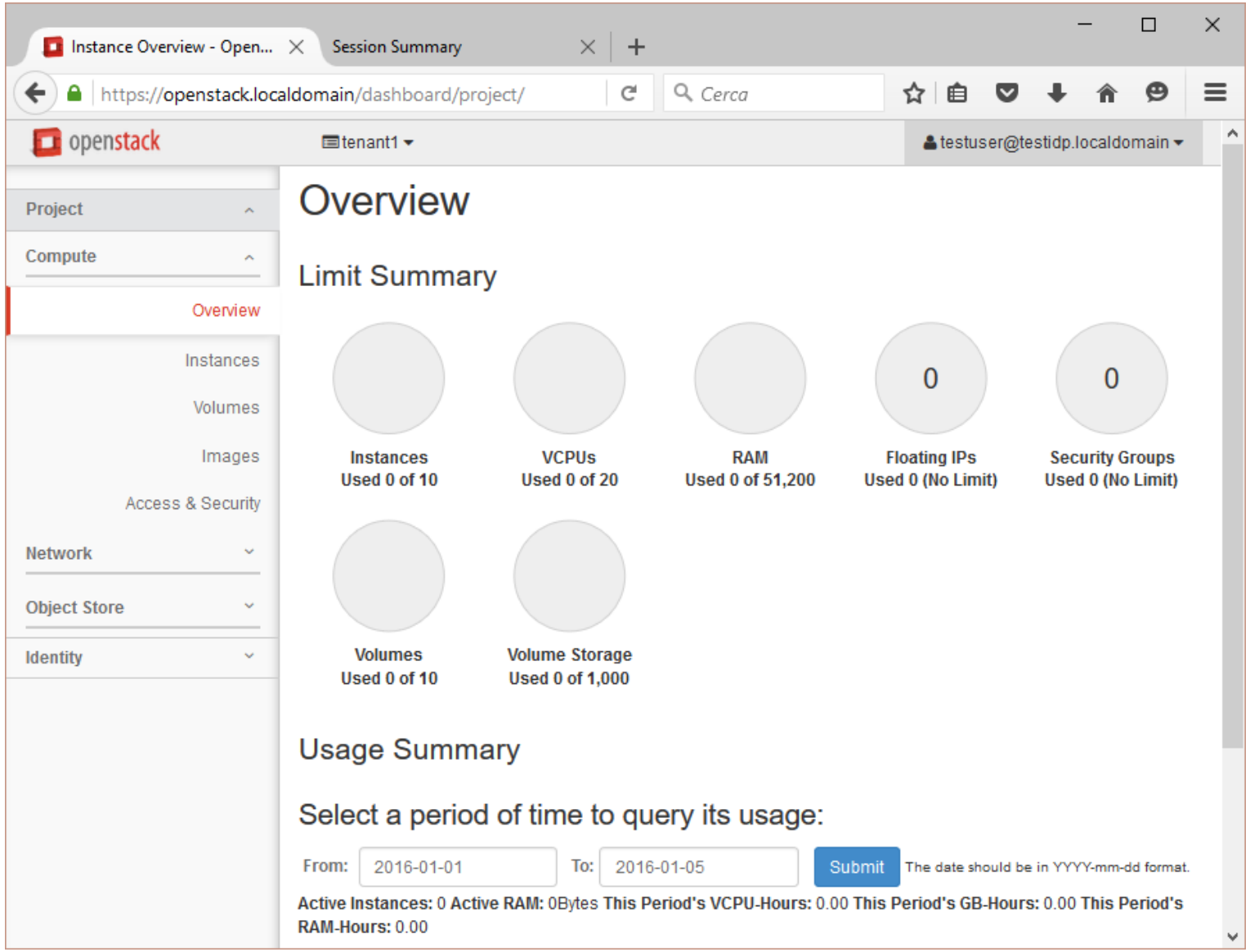
Live Demo

[2] OpenStack Registration Site



Live Demo

[3] Horizon interface with proper projects



Open points and future work

- Future work is necessary in the area of user de-provisioning.
  - currently, new users of the federation with the proper permissions can access the resources in OpenStack, and their corresponding user accounts are created as necessary in the system.
  - when they are no longer entitled to use the cloud, they will be denied from accessing those resources. This is achieved by the collaboration of Shibboleth, regsite and OpenStack.
  - however, the resources themselves will not be freed up, e.g. the virtual machines will not stop. This is a very complex task to solve.

- Another area of future work is enabling command line access (CLI).

Q & A