Keystone v3 and multi-backends

- Contacts on page 1
- Description on page 1
  - Architecture on page 2
- Usecase resolution on page 2
  - Setup LDAP on page 2
  - Enable Keystone V3 on page 4
  - Enable multi-backend on page 9
  - Dashboard Configuration on page 9
  - Link Keystone to LDAP on page 10
  - Link the openstack services to keystone V3 on page 11
  - Link external user to project on page 12
  - Testing on page 13
- Issues on page 13
- Useful links on page 14

This use-case was part of a OSP-D hackfest organized within the RCIP consultants team (January 2016).

Contacts

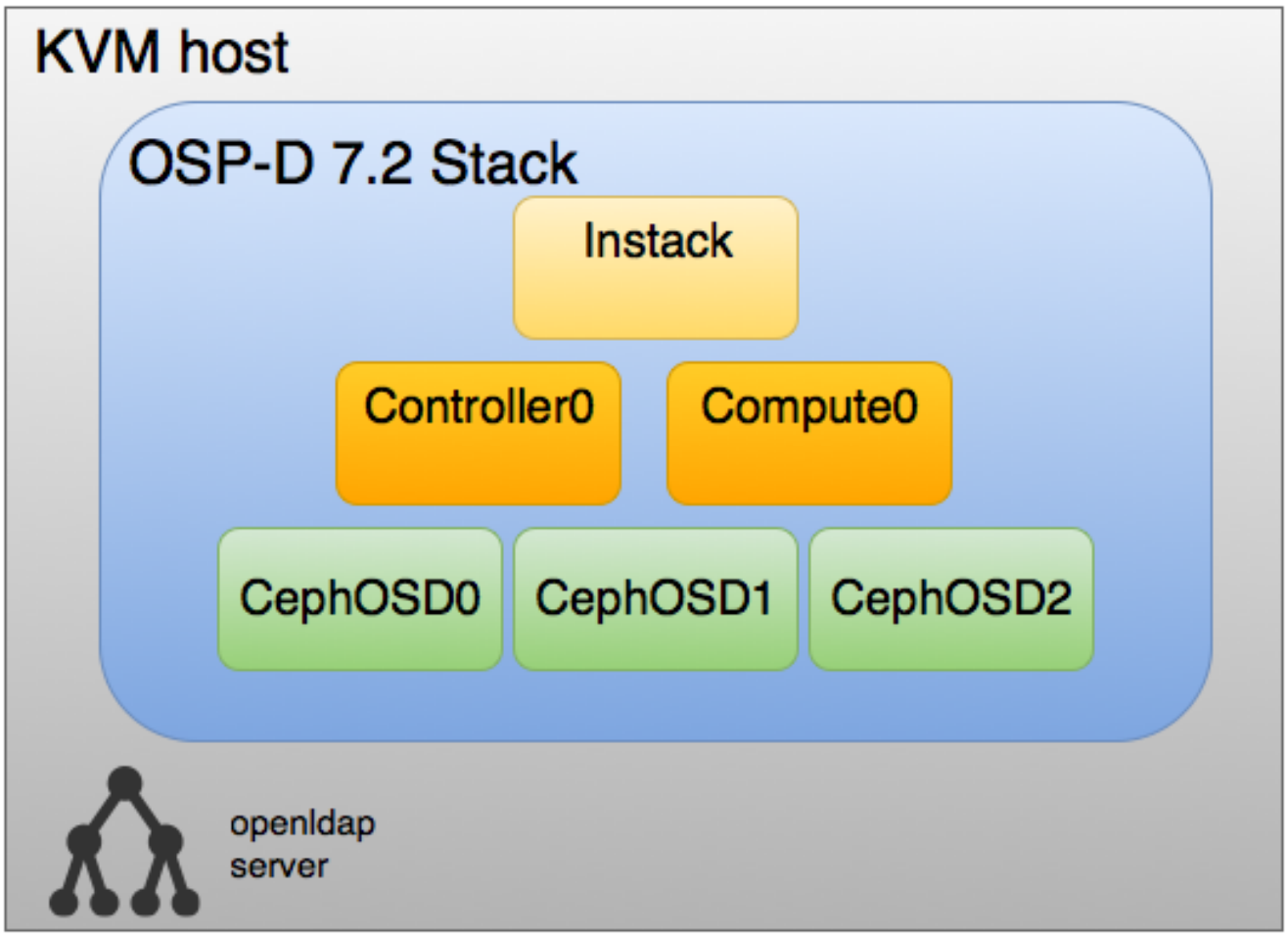
- Kevin Pichardie
- Fabien Cambi

Description

The goal of this use case is to take advantage of the multi-backend feature of keystone v3 with OSP-D 7.2.
At the end of this document you'll be able to authenticate your users through an openldap and affect them to specific projects.

The setup was done on a single KVM host.
The openldap server is on the KVM host.

Architecture



Usecase resolution

Setup LDAP

Install the packages

```
yum install openldap openldap-clients openldap-servers
```

Configure SELinux properly and start slapd

(or deactivate SElinux: setenforce 0)

```
setsebool -P authlogin_nsswitch_use_ldap=1
setsebool -P allow_ypbind=1
systemctl start slapd
```

Generate the encrypted password for "mypassword"

```
slappasswd -h {SSHA} -s mypassword
```

Create the provisionning files for a basic ldap tree

Be careful about space characters on the end of lines

Goal: Create the admin (Manager) access

File: manager.ldif

```
dn:  olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=redhat,dc=example,dc=com

replace: olcRootDN
olcRootDN: cn=Manager,dc=redhat,dc=example,dc=com

add: olcRootPW
olcRootPW: <Previously generated line from slappasswd>
```

Goal: Create the top tree "redhat.example.com", and subtrees "Groups", "Users" and "Roles".

File: structure.ldif

```
dn: dc=redhat,dc=example,dc=com
objectClass: dcObject
objectClass: organization
dc: redhat
o: redhat

dn: ou=Groups,dc=redhat,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups

dn: ou=Users,dc=redhat,dc=example,dc=com
objectClass: organizationalUnit
ou: Users

dn: ou=Roles,dc=redhat,dc=example,dc=com
objectClass: organizationalUnit
ou: Roles
```

Goal: Create the groups "sysadmins" and "marketing".

File: add_groups.ldif

```
dn: cn=sysadmins,ou=Groups,dc=redhat,dc=example,dc=com
cn: sysadmin
objectClass: top
objectClass: groupofnames
member: cn=userA,ou=people,dc=redhat,dc=example,dc=com
description: UNIX systems administrators
```

```
dn: cn=marketing,ou=Groups,dc=redhat,dc=example,dc=com
cn: marketing
objectClass: top
objectClass: groupofnames
member: cn=userB,ou=people,dc=redhat,dc=example,dc=com
description: Marketing
```

Goal: Create the users userA and userB.

File: add_users.ldif

```
dn: cn=userA,ou=Users,dc=redhat,dc=example,dc=com
cn: userA
sn: User A
objectClass: top
objectClass: person
#pass: usera
userPassword: {SSHA}qFw52ZHBxEb3AnsiniDPGuCO0wgBNhFG

dn: cn=userB,ou=Users,dc=redhat,dc=example,dc=com
cn: userB
sn: User B
objectClass: top
objectClass: person
#pass: userb
userPassword: {SSHA}S9qx0iTJ+8bLLjwn/gL0/Qdl9dBPKk+B
```

Provision the ldap tree

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f manager.ldif
ldapadd -x -D "cn=Manager,dc=redhat,dc=example,dc=com" -H ldap://localhost  -w mypassword -
f structure.ldif
ldapadd -x -D "cn=Manager,dc=redhat,dc=example,dc=com" -H ldap://localhost  -w mypassword -
f add_groups.ldif
ldapadd -x -D "cn=Manager,dc=redhat,dc=example,dc=com" -H ldap://localhost  -w mypassword -
f add_users.ldif
```

Test the ldap

```
ldapsearch -x -w mypassword -D "cn=Manager,dc=redhat,dc=example,dc=com" -P
"dc=redhat,dc=example,dc=com"
```

This command should output the entine ldap records starting from the top tree "example.com".

Enable Keystone V3

Prerequisites

- Deploy a typical 5 nodes Openstack platform using OSP-d 7.2
- SSH to your overcloud controller(s)

Information needed from post-deployment

Export default variables :

```
export HOSTAUTH=$(keystone endpoint-list | grep "dashboard" | awk '{print $6}' | cut -d"/"
-f3 | cut -d":" -f1)
export MYREGION="regionOne"
export ADMIN_TOKEN=$(sudo grep ^admin_token /etc/keystone/keystone.conf | awk '{print
$3}')" source overcloudrc
```

Create V3 service endpoint

Create a identityv3 service :

```
keystone service-create --name keystonev3 --type identityv3 --description "Keystone
Identity Service v3"
```

Create the apropiate endpoints, if you use SSL please change http for https :

```
keystone endpoint-create --region $MYREGION --service keystonev3 --publicurl "http://
$HOSTAUTH:5000/v3" --adminurl "http://$HOSTAUTH:35357/v3" --internalurl "http://
$HOSTAUTH:5000/v3"
```

Create basics domains on V3 api

Openstack cli isn't fully compliant with osp 7.2 api version so we will set all V3 request througth direct api.

Get the admin token:

```
ADMIN_TOKEN=$(\
curl http://$HOSTAUTH:5000/v3/auth/tokens \
    -s \
     -i \
      -H "Content-Type: application/json" \
       -d '
    {
        "auth": {
             "identity": {
                  "methods": [
                       "password"
                   ],
                    "password": {
                         "user": {
                              "domain": {
                                   "name": "Default"
                                },
                                "name": "admin",
                                "password": "'${OS_PASSWORD}'"
                           }
                     }
                },
                "scope": {
```

```
                        "project": {
                             "domain": {
                                  "name": "Default"
                             },
                         "name": "admin"
                          }
                }
            }
      }' | grep ^X-Subject-Token: | awk '{print $2}' )
```

Create a new admin_domain:

```
ID_ADMIN_DOMAIN=$(\
curl http://$HOSTAUTH:5000/v3/domains \
  -s  \
  -H "X-Auth-Token: $ADMIN_TOKEN" \
  -H "Content-Type: application/json"  -d '{   "domain": { "enabled": true,   "name":
"admin_domain" }}'  | jq .domain.id | tr -d '"' )
```

Create a cloud admin user:

```
ID_CLOUD_ADMIN=$(\
curl http://$HOSTAUTH:5000/v3/users   -s   -H "X-Auth-Token: $ADMIN_TOKEN" -H "Content-
Type: application/json" -d "{
       \"user\": {
           \"description\": \"Cloud administrator\",
           \"domain_id\": \"$ID_ADMIN_DOMAIN\",
           \"enabled\": true,
           \"name\": \"cloud_admin\",
           \"password\": \"password\"
       }
}" | jq .user.id | tr -d '"' )
```

Get the admin role ID:

```
ADMIN_ROLE_ID=$(\
curl http://$HOSTAUTH:5000/v3/roles?name=admin \
    -s \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
| jq .roles[0].id | tr -d '"' )
```

Add user cloud_admin as admin:

```
curl -X PUT http://$HOSTAUTH:5000/v3/domains/${ID_ADMIN_DOMAIN}/users/${ID_CLOUD_ADMIN}/
roles/${ADMIN_ROLE_ID} \
    -s \
    -i \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
    -H "Content-Type: application/json
```

Check cloud-admin role:

```
curl http://$HOSTAUTH:5000/v3/domains/${ID_ADMIN_DOMAIN}/users/${ID_CLOUD_ADMIN}/roles \
    -s \
    -H "X-Auth-Token: $ADMIN_TOKEN" | jq .roles
```

Get cloud-admin token:

```
CLOUD_ADMIN_TOKEN=$(\
curl http://$HOSTAUTH:5000/v3/auth/tokens  -s  -i -H "Content-Type: application/json"  -d '
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "domain": {
                         "name": "admin_domain"
                     },
                     "name": "cloud_admin",
                     "password": "password"
                 }
            }
        },
        "scope": {
             "domain": {
                 "name": "admin_domain"
             }
        }
    }
}' | grep ^X-Subject-Token: | awk '{print $2}' )
```

Create a new domain for multi domain:

```
ID_DOM1=$(\
curl http://$HOSTAUTH:5000/v3/domains \
-s \
-H "X-Auth-Token: $CLOUD_ADMIN_TOKEN" \
-H "Content-Type: application/json" \
-d '{
    "domain": {
         "enabled": true,
         "name": "dom1"
     }
}' | jq .domain.id | tr -d '"')

echo "ID of dom1: $ID_DOM1"
```

Create admin for this domain:

```
ID_ADM1=$(\
curl http://$HOSTAUTH:5000/v3/users \
-s \
-H "X-Auth-Token: $CLOUD_ADMIN_TOKEN" \
-H "Content-Type: application/json" \
-d "{
    \"user\": {
```

```
        \"description\": \"Administrator of domain dom1\",
        \"domain_id\": \"$ID_DOM1\",
        \"enabled\": true,
        \"name\": \"adm1\",
        \"password\": \"password\"
    }
}" | jq .user.id | tr -d '"')

echo "ID of user adm1: $ID_ADM1"
```

Set role admin:

```
curl -X PUT http://$HOSTAUTH:5000/v3/domains/${ID_DOM1}/users/${ID_ADM1}/roles/
${ADMIN_ROLE_ID} \
    -s \
    -i \
    -H "X-Auth-Token: $CLOUD_ADMIN_TOKEN" \
    -H "Content-Type: application/json"
```

Check role for this user:

```
curl http://$HOSTAUTH:5000/v3/domains/${ID_DOM1}/users/${ID_ADM1}/roles \
    -s \
    -H "X-Auth-Token: $CLOUD_ADMIN_TOKEN" | jq .roles
```

Add a new project:

```
ID_PRJ1=$(\
curl http://$HOSTAUTH:5000/v3/projects \
    -s \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
    -H "Content-Type: application/json" \
    -d "
{
    \"project\": {
        \"enabled\": true,
        \"domain_id\": \"$ID_DOM1\",
        \"name\": \"prj1\"
    }\
}" | jq .project.id | tr -d '"' )

echo "ID of prj1: $ID_PRJ1"
```

Set policy v3 compliant for controller nodes:

```
wget https://github.com/openstack/keystone/blob/master/etc/policy.v3cloudsample.json

sed s/admin_domain_id/${ID_ADMIN_DOMAIN}/ \
    < /etc/keystone/policy.v3cloudsample.json \
    > /etc/keystone/policy.json-v3

cp /etc/keystone/policy.json /etc/keystone/policy.json-v2
cp /etc/keystone/policy.json-v3 /etc/keystone/policy.json
chown keystone /etc/keystone/policy.json
```

Restart keystone api service:

```
systemctl restart openstack-keystone.service
```

Enable multi-backend

Configure Identity Service to use multiple back ends

```
mkdir /etc/keystone/domains/
chown keystone /etc/keystone/domains/

# openstack-config --set /etc/keystone/keystone.conf identity
domain_specific_drivers_enabled true
# openstack-config --set /etc/keystone/keystone.conf identity domain_config_dir /etc/
keystone/domains
# openstack-config --set /etc/keystone/keystone.conf assignment driver
keystone.assignment.backends.sql.Assignment
```

If cli isn't working, like on 7.2 for example just replace the value in /etc/keystone/keystone.conf

Restart keystone to enable multi backends:

```
systemctl restart openstack-keystone.service
```

Dashboard Configuration

Dashboard configuration for Keystone V3

To enable dashboard functionning with Keystone V3, on all controller nodes you need to edit /etc/openstack-dashboard/local_settings and change those lines:/!\ wrong indentation in the above file will result in apache not starting /!\

```
OPENSTACK_API_VERSIONS = {
"identity": 3,
}
[...]
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
[...]
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
[...]
OPENSTACK_KEYSTONE_URL = "http://[API VIP IP]:5000/v3
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"
[...]
#SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
```

Then you need to restart httpd:

```
systemctl restart httpd
```

Link Keystone to LDAP

Create LDAP domain in keystone:

```
ID_DOMRH=$(\
curl http://$HOSTAUTH:5000/v3/domains \
-s \
-H "X-Auth-Token: $CLOUD_ADMIN_TOKEN" \
-H "Content-Type: application/json" \
-d '{
    "domain": {
         "enabled": true,
         "name": "redhat"
     }
}' | jq .domain.id | tr -d '"')

echo "ID of domain rh: $ID_DOMRH"
```

Create the domain configuration file:

```
touch /etc/keystone/domains/keystone.[domain name in keystone].conf
chown keystone -R /etc/keystone/domains/
# This file should match the domain created in keystone. So you'll have to replace the
values with your information.
```

Configuration of ldap keystone backend:

File: /etc/keystone/domains/keystone.redhat.conf

```
[ldap]
#IP of the openldap server
url =  ldap://[openldap server ip]/

#User used to manage the ldap (should match olcRootDN from manager.ldif file)
user = cn=Manager,dc=redhat,dc=example,dc=com

#Password (should match olcRootPW from manager.ldif file)
password              = olcRootPW

# Suffix of the ldap (should match olcSuffix from manager.ldif file)
suffix                = dc=redhat,dc=example,dc=com

# Values of the ldap attributes of a user (should match the information of the file
add_users.ldif)
user_objectclass     = person
user_id_attribute    = cn
user_name_attribute  = cn

#Uncomment if you use ssl
#use_tls = False

[identity]
driver = keystone.identity.backends.ldap.Identity
```

Restart keystone to enable multi backends:

```
systemctl restart openstack-keystone.service
```

Link the openstack services to keystone V3

On the controllers you need to edit the configuration of each service.

Nova configuration

File: **/etc/nova/nova.conf**

```
auth_uri=http://[api-vip-ip]:5000/v3
auth_version=v3
systemctl restart openstack-nova-api.service
```

Neutron configuration

File: **/etc/neutron/neutron.conf**

```
auth_uri=http://[api-vip-ip]:5000/v3
systemctl restart neutron-server.service
```

Cinder configuration

File: **/etc/cinder/api-paste.ini**

```
auth_uri=http://[api-vip-ip]:5000/v3
systemctl restart openstack-cinder-api.service
```

Ceilometer configuration

File: **/etc/ceilometer/ceilometer.conf**

```
auth_uri=http://[api-vip-ip]:5000/v3
systemctl restart openstack-ceilometer-api.service openstack-ceilometer-central.service
```

Glance configuration

File: **/etc/glance/glance-api.conf**

and

File: **/etc/glance/glance-registry.conf**

```
auth_uri=http://[api-vip-ip]:5000/v3
systemctl restart openstack-glance-api.service
```

Heat configuration

File: **/etc/heat/heat.conf**

```
auth_uri=http://[api-vip-ip]:5000/v3
auth_version=v3
systemctl restart openstack-heat-api.service  openstack-heat-api-cloudwatch.service
openstack-heat-api-cfn.service
```

Link external user to project

Get the ldap user id for user "username":

```
USER_LDAP_ID=$(\
curl http://$HOSTAUTH:5000/v3/users?domain_id=$ID_DOMRH     -s     -H "X-Auth-Token:
$ADMIN_TOKEN"  | jq  '.users[] | select(.name=="username") | .id' | tr -d '"')
```

Get the member role id:

```
MEMBER_ROLE_ID=$(\
curl http://$HOSTAUTH:5000/v3/roles?name=_member_ \
    -s \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
| jq .roles[0].id | tr -d '"' )
```

Add a new project:

```
ID_PRJRH=$(\
curl http://$HOSTAUTH:5000/v3/projects \
    -s \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
    -H "Content-Type: application/json" \
    -d "
{
    \"project\": {
        \"enabled\": true,
        \"domain_id\": \"$ID_DOMRH\",
        \"name\": \"prjrh\"
    }\
}" | jq .project.id | tr -d '"' )

echo "ID of prjrh: $ID_PRJRH"
```

Add role member to an external user on a project :

```
curl -X PUT http://$HOSTAUTH:5000/v3/projects/${ID_PRJRH}/users/${USER_LDAP_ID}/roles/
${MEMBER_ROLE_ID} \
    -s \
    -i \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
    -H "Content-Type: application/json"
```

Now this user can connect and start using Openstack.

Testing

Setup an SSH tunnel, connect to the dashboard and see if you can connect using the credentials you created:
login: <username>
pass: <password>
domain: redhat

```
curl -X PUT http://$HOSTAUTH:5000/v3/projects/${ID_PRJRH}/users/${USER_LDAP_ID}/roles/
${MEMBER_ROLE_ID} \
    -s \
    -i \
    -H "X-Auth-Token: $ADMIN_TOKEN" \
    -H "Content-Type: application/json"
```

Issues

If you can't login, on the controller you can look at the keystone logs:

```
sudo -i
tail -f /var/log/keystone/keystone.log
```

You should see something like:

```
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 fd=11 ACCEPT from
IP=192.168.122.21:52875 (IP=0.0.0.0:389)
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=0 BIND
dn="cn=Manager,dc=redhat,dc=example,dc=com" method=128
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=0 BIND
dn="cn=Manager,dc=redhat,dc=example,dc=com" mech=SIMPLE ssf=0
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=0 RESULT tag=97 err=0 text=
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=1 SRCH
base="ou=Users,dc=redhat,dc=example,dc=com" scope=1 deref=0 filter="(&(cn=fcambi)
(objectClass=person))"
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=1 SRCH attr=mail userPassword
enabled cn
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=1 SEARCH RESULT tag=101 err=0
nentries=1 text=
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 op=2 UNBIND
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1279 fd=11 closed
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 fd=11 ACCEPT from
IP=192.168.122.21:52876 (IP=0.0.0.0:389)
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=0 BIND
dn="cn=Manager,dc=redhat,dc=example,dc=com" method=128
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=0 BIND
dn="cn=Manager,dc=redhat,dc=example,dc=com" mech=SIMPLE ssf=0
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=0 RESULT tag=97 err=0 text=
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=1 SRCH
base="ou=UserGroups,dc=redhat,dc=example,dc=com" scope=1 deref=0
```

```
filter="(&(&(objectClass=groupOfNames)
(member=cn=fcambi,ou=users,dc=redhat,dc=example,dc=com))(objectClass=groupOfNames))"
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=1 SRCH attr=ou cn description
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=1 SEARCH RESULT tag=101 err=32
nentries=0 text=
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 op=2 UNBIND
jan 18 23:24:29 hackfest-node2 slapd[717]: conn=1280 fd=11 closed
```

If come components (i.e heat) seem to be unresponsive, try to check the status of pacemaker and clean the resources:

```
pcs status
pcs resource cleanup <failed resource>
```

Useful links

https://access.redhat.com/articles/1270363
https://access.redhat.com/articles/1193253
https://access.redhat.com/node/1407823
https://access.redhat.com/node/1339493
https://access.redhat.com/documentation/en/red-hat-enterprise-linux-openstack-platform/7/integrate-with-identity-service/chapter-1-active-directory-integration
https://access.redhat.com/node/2019453
https://wiki.openstack.org/wiki/Horizon/DomainWorkFlow
http://www.florentflament.com/blog/setting-keystone-v3-domains.html
http://adam.younglogic.com/2015/05/rdo-v3-only/