OpenStack: How can I integrate Bright OpenStack 7.1 Keystone with LDAP/AD?

How can I integrate Keystone with LDAP?

Here it is assumed that OpenStack was set up by Bright Cluster Manager 7.1 and is up and running.

The following steps describe the procedure needed to configure Keystone to use LDAP as a backend for user authentication, instead of MySQL.

The OpenLDAP that Bright Cluster Manager provides is used here, but the instructions can be made to work with any other standard LDAP server.

The end result is a Keystone deployment which authenticates users against LDAP. But it still uses its local MySQL database for authorization, storing info on roles, projects, assignments, etc.

Credentials Retrieval From Keystone

First, disable the CMDaemon integration with OpenStack with the following commands using `cmsh` on the head node:

```
$ cmsh -c 'openstack use default; set enabled 0; commit'
```

Next, get the currently configured usernames/passwords for OpenStack service. Since we want keystone to authenticate against LDAP, this will also include authentication of OpenStack services.

The `cmsh` shell running on the head node can retrieve these pairs as follows:

```
$ cmsh -c 'openstack settingscredentials; get cinderusername; get cinderpassword'
```

```
$ cmsh -c 'openstack settingscredentials; get keystoneusername; get keystonepassword'

$ cmsh -c 'openstack settingscredentials; get cmdaemonopenstackusername; get cmdaemonopenstackpassword'

$ cmsh -c 'openstack settingscredentials; get glanceusername; get glancepassword'

$ cmsh -c 'openstack settingscredentials; get heatusername; get heatpassword'

$ cmsh -c 'openstack settingscredentials; get neutronusername; get neutronpassword'

$ cmsh -c 'openstack settingscredentials; get novausername; get novapassword'

$ cmsh -c 'openstack settingscredentials; get userec2username; get userec2password'

$ echo admin && cmsh -c 'openstack settingscredentials; get mainadminpassword'
```

User creation

Next step is to create OpenStack System users in LDAP using the credentials from the previous step. The users can be created by hand using `cmgui` or `cmsh` (Chapter 6 - User Management, Bright Cluster Manager administration manual. [1]).

By default you will want to recreate all of the OpenStack system users in LDAP in the same way they were before you started the process (users nova, glance, keystone, etc.). However, in principle it is also possible to create just one (shared) user for all the services (eg. cmd) and the a separate admin user. This is handy in case of name conflicts in LDAP (e.g. a user 'nova' already existing), or if for some reason administrator wants to limit the number of additional accounts in LDAP. If that's what you will want to do, after determining the new username(s) for OpenStack services, and after creating those accounts in LDAP, you will have to upated the credentials for them services in CMDaemon. This can be done with `cmgui` in the Openstack menu' -> Settings -> Credentials settings, or it can be done with `cmsh` as follows:

```
$ cmsh -c 'openstack settingscredentials; set cinderusername <USER>; set cinderpassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set keystoneusername <USER>; set keystonepassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set cmdaemonopenstackusername <USER>; set cmdaemonopenstackpassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set glanceusername <USER>; set glancepassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set heatusername <USER>; set heatpassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set neutronusername <USER>; set neutronpassword <PASSWORD>; commit'
```

```
$ cmsh -c 'openstack settingscredentials; set novausername <USER>; set novapassword <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set userec2username <USER>; set userec2password <PASSWORD>; commit'

$ cmsh -c 'openstack settingscredentials; set mainadminpassword <PASSWORD>; commit'
```

OpenStack services can then be enabled again with the following command:

```
$ cmsh -c 'openstack use default; set enabled 1; commit'
```

Modifying Keystone To Use LDAP

The Keystone configuration must now be modified to use the LDAP driver for the identity backend, and the MySQL driver for the Assignment backend.

The password, the searchdn and the readonlyuser are extracted with the following command from the headnode, because they are going to be needed soon:

```
$ cat /cm/local/apps/cmd/etc/cmd.conf | grep -E 'LDAPReadOnlyUser|LDAPReadOnlyPass|LDAPSearchDN'
```

The file '`/etc/keystone/keystone.conf`' is modified in the `ldap` section, using the `crudini` command for Keystone[2]. This should also be done on any Keystone server running on the passive:

```
$ crudini --set /etc/keystone/keystone.conf ldap url ldap://master

$ crudini --set /etc/keystone/keystone.conf ldap user cn=readonlyroot,dc=cm,dc=cluster

$ crudini --set /etc/keystone/keystone.conf ldap password <PASSWORD>

$ crudini --set /etc/keystone/keystone.conf ldap suffix dc=cm,dc=cluster
```

```
$ crudini --set /etc/keystone/keystone.conf ldap use_dumb_member false

$ crudini --set /etc/keystone/keystone.conf ldap allow_subtree_delete false

$ crudini --set /etc/keystone/keystone.conf ldap user_tree_dn dc=cm,dc=cluster

$ crudini --set /etc/keystone/keystone.conf ldap user_objectclass inetOrgPerson

$ crudini --set /etc/keystone/keystone.conf ldap user_id_attribute entryUUID

$ crudini --set /etc/keystone/keystone.conf ldap user_name_attribute uid

$ crudini --set /etc/keystone/keystone.conf ldap user_mail_attribute mail

$ crudini --set /etc/keystone/keystone.conf ldap user_pass_attribute userPassword

$ crudini --set /etc/keystone/keystone.conf ldap user_id_attribute entryUUID

$ crudini --set /etc/keystone/keystone.conf ldap user_allow_create false

$ crudini --set /etc/keystone/keystone.conf ldap user_allow_update false

$ crudini --set /etc/keystone/keystone.conf ldap user_allow_delete false

$ crudini --set /etc/keystone/keystone.conf ldap group_tree_dn ou=Group,dc=cm,dc=cluster

$ crudini --set /etc/keystone/keystone.conf ldap group_objectclass posixGroup

$ crudini --set /etc/keystone/keystone.conf ldap group_id_attribute entryUUID

$ crudini --set /etc/keystone/keystone.conf ldap group_name_attribute cn

$ crudini --set /etc/keystone/keystone.conf ldap group_allow_create false

$ crudini --set /etc/keystone/keystone.conf ldap group_allow_update false
```

```
$ crudini --set /etc/keystone/keystone.conf ldap group_allow_delete false
```

For other LDAP servers the key/value pairs may differ. Two important keys are `user_id_attribute` and `group_id_attribute` -- they **must** be mapped to unique values.

In the identity section of keystone.conf, the LDAP backend driver must be set with `crudini` or `cmsh` commands:

```
$ crudini --set /etc/keystone/keystone.conf identity driver keystone.identity.backends.ldap.Identity
```

In the assignment section of `keystone.conf`, the SQL backend driver must be set with the `crudini` or `cmsh` commands:

```
$ crudini --set /etc/keystone/keystone.conf assignment driver keystone.assignment.backends.sql.Assignment
```

Keystone must then be restarted on the head node:

```
$ service openstack-keystone restart
```

Getting A Token For The Admin User, And Assigning Roles

The token for the admin user is set using the following commands. The token is needed for the next step. Wait some seconds before running the `grep` command:

```
$ cmsh -c 'openstack; settingscredentials; set admintoken `openssl rand -hex 10`; commit'

$ grep admin_token= /etc/keystone/keystone.conf | grep -vE '^#'
```

The role is assigned to the admin and services users as follows (Replace `<TOKEN>` with the token that was just grepped):

```
$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin  --project bright admin
```

```
$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon  --project bright admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user glance --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user glance --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user keystone --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user keystone --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user nova --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user nova --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user neutron --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user neutron --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cinder --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cinder --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user heat --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user heat --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon --project service member
```

If there is a single user for all the services then run the following command instead (Replace `<TOKEN>` with the grepped token from earlier):

```
$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin --project bright admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon --project bright admin
```

OpenStack: How can I integrate Bright OpenStack 7.1 Keystone with LDAP/AD?

```
$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user cmdaemon --project service member

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin --project service admin

$ openstack --os-token <TOKEN> --os-url http://master:5000/v3 role add --user admin --project service member
```

The configuration is now complete. It is now possible to login with the admin user on the OpenStack dashboard, and assign a role to a user created by Bright Cluster Manager in the OpenLDAP.

That is: It is now possible to, for example, create a user in LDAP/AD, and then immediately log in as that user to Keystone.

[1] http://support.brightcomputing.com/manuals/7.0/admin-manual.pdf

[2] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/5/html/Cloud_Administrator_Guide/configuring-keystone-for-ldap-backend.html

Unique solution ID: #1273
Author: Matteo Piccinini
Last update: 2015-06-18 11:01