INTEGRATING WITH OPENSTACK KEYSTONE

It is possible to integrate the Ceph Object Gateway with Keystone, the OpenStack identity service. This sets up the gateway to accept Keystone as the users authority. A user that Keystone authorizes to access the gateway will also be automatically created on the Ceph Object Gateway (if didn’t exist beforehand). A token that Keystone validates will be considered as valid by the gateway.

The following configuration options are available for Keystone integration:

```
[client.radosgw.gateway]
rgw keystone api version = {keystone api version}
rgw keystone url = {keystone server url:keystone server admin port}
rgw keystone admin token = {keystone admin token}
rgw keystone accepted roles = {accepted user roles}
rgw keystone token cache size = {number of tokens to cache}
rgw keystone revocation interval = {number of seconds before checking revoked tickets}
rgw keystone implicit tenants = {true for private tenant for each new user}
rgw s3 auth use keystone = true
nss db path = {path to nss db}
```

It is also possible to configure a Keystone service tenant, user & password for keystone (for v2.0 version of the OpenStack Identity API), similar to the way OpenStack services tend to be configured, this avoids the need for setting the shared secret `rgw keystone admin token` in the configuration file, which is recommended to be disabled in production environments. The service tenant credentials should have admin privileges, for more details refer the Openstack keystone documentation, which explains the process in detail. The requisite configuration options for are:

```
rgw keystone admin user = {keystone service tenant user name}
rgw keystone admin password = {keystone service tenant user password}
rgw keystone admin tenant = {keystone service tenant name}
```

A Ceph Object Gateway user is mapped into a Keystone `tenant`. A Keystone user has different roles assigned to it on possibly more than a single tenant. When the Ceph Object Gateway gets the ticket, it looks at the tenant, and the user roles that are assigned to that ticket, and accepts/rejects the request according to the `rgw keystone accepted roles` configurable.

For a v3 version of the OpenStack Identity API you should replace `rgw keystone admin tenant` with:

```
rgw keystone admin domain = {keystone admin domain name}
rgw keystone admin project = {keystone admin project name}
```

PRIOR TO KILO

Keystone itself needs to be configured to point to the Ceph Object Gateway as an object-storage endpoint:

```
keystone service-create --name swift --type object-store
keystone endpoint-create --service-id <id> --publicurl http://radosgw.example.com/swift/v1 \
        --internalurl http://radosgw.example.com/swift/v1 --adminurl http://radosgw.example.c
```

AS OF KILO

Keystone itself needs to be configured to point to the Ceph Object Gateway as an object-storage endpoint:

```
openstack service create --name=swift \
                         --description="Swift Service" \
                         object-store
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | Swift Service                    |
| enabled     | True                             |
| id          | 37c4c0e79571404cb4644201a4a6e5ee |
| name        | swift                            |
| type        | object-store                     |
+-------------+----------------------------------+

openstack endpoint create --region RegionOne \
     --publicurl   "http://radosgw.example.com:8080/swift/v1" \
     --adminurl    "http://radosgw.example.com:8080/swift/v1" \
     --internalurl "http://radosgw.example.com:8080/swift/v1" \
     swift
+--------------+------------------------------------------+
| Field        | Value                                    |
+--------------+------------------------------------------+
| adminurl     | http://radosgw.example.com:8080/swift/v1 |
| id           | e4249d2b60e44743a67b5e5b38c18dd3         |
| internalurl  | http://radosgw.example.com:8080/swift/v1 |
| publicurl    | http://radosgw.example.com:8080/swift/v1 |
| region       | RegionOne                                |
| service_id   | 37c4c0e79571404cb4644201a4a6e5ee         |
| service_name | swift                                    |
| service_type | object-store                             |
+--------------+------------------------------------------+

$ openstack endpoint show object-store
+--------------+------------------------------------------+
| Field        | Value                                    |
+--------------+------------------------------------------+
| adminurl     | http://radosgw.example.com:8080/swift/v1 |
| enabled      | True                                     |
| id           | e4249d2b60e44743a67b5e5b38c18dd3         |
| internalurl  | http://radosgw.example.com:8080/swift/v1 |
| publicurl    | http://radosgw.example.com:8080/swift/v1 |
| region       | RegionOne                                |
| service_id   | 37c4c0e79571404cb4644201a4a6e5ee         |
| service_name | swift                                    |
| service_type | object-store                             |
+--------------+------------------------------------------+
```

The keystone URL is the Keystone admin RESTful API URL. The admin token is the token that is configured internally in Keystone for admin requests.

The Ceph Object Gateway will query Keystone periodically for a list of revoked tokens. These requests are encoded and signed. Also, Keystone may be configured to provide self-signed tokens, which are also encoded and signed. The gateway needs to be able to decode and verify these signed messages, and the process requires that the gateway be set up appropriately. Currently, the Ceph Object Gateway will only be able to perform the procedure if it was compiled with `--with-nss`. Configuring the Ceph Object Gateway to work with Keystone also requires converting the OpenSSL certificates that Keystone uses for creating the requests to the nss db format, for example:

```
mkdir /var/ceph/nss

openssl x509 -in /etc/keystone/ssl/certs/ca.pem -pubkey | \
        certutil -d /var/ceph/nss -A -n ca -t "TCu,Cu,Tuw"
openssl x509 -in /etc/keystone/ssl/certs/signing_cert.pem -pubkey | \
        certutil -A -d /var/ceph/nss -n signing_cert -t "P,P,P"
```

Openstack keystone may also be terminated with a self signed ssl certificate, in order for radosgw to interact with keystone in such a case, you could either install keystone’s ssl certificate in the node running radosgw. Alternatively radosgw could be made to not verify the ssl certificate at all (similar to openstack clients with a `--insecure` switch) by setting the value of the configurable `rgw keystone verify ssl` to false.