Red Hat Ceph Storage 3

Using Keystone to Authenticate Ceph Object Gateway Users

Configuring OpenStack and the Ceph Object Gateway to use Keystone for user authentication.

Last Updated: 2021-09-08

Red Hat Ceph Storage 3 Using Keystone to Authenticate Ceph Object Gateway Users

Configuring OpenStack and the Ceph Object Gateway to use Keystone for user authentication.

Legal Notice

Abstract

This document describes how to configure OpenStack and the Ceph Object Gateway to use Keystone for user authentication.

Table of Contents

PREFACE

Organizations using OpenStack Keystone to authenticate users can integrate Keystone with the Ceph Object Gateway, which enables the gateway to accept a Keystone token, authenticate the user and create a corresponding Ceph Object Gateway user. When Keystone validates a token, the gateway considers the user authenticated.

Benefits include:

- Managing Users with Keystone
- Automatic User Creation in the Ceph Object Gateway
- The Ceph Object Gateway will query Keystone periodically for a list of revoked tokens.

CHAPTER 1. CONFIGURING OPENSTACK

Before configuring the Ceph Object Gateway, configure Keystone so that the Swift service is enabled and pointing to the Ceph Object Gateway.

1.1. CREATING THE SWIFT SERVICE

To use OpenStack to validate Swift users, first create the Swift service.

```
# openstack service create --name=swift --description="Swift Service" object-store
```

Creating the service will echo the service settings. For example:

Field	Value
description	Swift Service
enabled	True
id	37c4c0e79571404cb4644201a4a6e5ee
name	swift
type	object-store

1.2. SETTING THE ENDPOINTS

After creating the Swift service, point it to a Ceph Object Gateway. Replace **{region-name}** with the name of the gateway’s zone group name or region name. Replace the exemplary URLs with URLs appropriate for the Ceph Object Gateway.

```
# openstack endpoint create --region {region-name} \
    --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
```

Setting the endpoints will echo the service endpoint settings. For example:

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

Field	Value
region	us-west
service_id	37c4c0e79571404cb4644201a4a6e5ee
service_name	swift
service_type	object-store

1.3. VERIFYING THE SETTINGS

After creating the Swift service and setting the endpoints, show the endpoints to ensure that all the settings are correct.

```
# openstack endpoint show object-store
```

Showing the endpoints will echo the endpoints settings, and the service settings. For example:

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	us-west
service_id	37c4c0e79571404cb4644201a4a6e5ee
service_name	swift
service_type	object-store

CHAPTER 2. CONFIGURING THE CEPH OBJECT GATEWAY

2.1. CONFIGURING SSL

Configuring the Ceph Object Gateway to work with Keystone 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 the Ceph Object Gateway to interact with Keystone. Either install Keystone's SSL certificate in the node running the Ceph Object Gateway, or alternatively set the value of the configurable **rgw_keystone_verify_ssl** setting to **false**. Setting **rgw_keystone_verify_ssl** to **false** means that the gateway won't attempt to verify the certificate.

2.2. CONFIGURING CIVETWEB

To configure the Ceph Object Gateway to use Keystone, open the Ceph configuration file on the admin node and navigate to the **[client.radosgw.{instance-name}]**, where **{instance-name}** is the name of the Gateway instance to configure. For each gateway instance, set the **rgw_s3_auth_use_keystone** setting to **true**, and set the **nss_db_path** setting to the path where the NSS database is stored.

Provide authentication credentials. It is possible to configure a Keystone service tenant, user and password for keystone for v2.0 version of the OpenStack Identity API, similar to the way system administrators tend to configure OpenStack services. Providing a username and password avoids providing the shared secret to the **rgw_keystone_admin_token** setting. Red Hat recommends disabling authentication by admin token in production environments.

The service tenant credentials should have **admin** privileges. For more details see the *Users and Identity Management Guide* for Red Hat OpenStack Platform 13. 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.

A typical configuration might have the following settings:

```
[client.radosgw.gateway]
rgw_keystone_url = {keystone server url:keystone server admin port}
##Authentication using an admin token. Not preferred.
#rgw_keystone_admin_token = {keystone admin token}
##Authentication using username, password and tenant. Preferred.
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}
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_make_new_tenants = {true for private tenant for each new user}
rgw_s3_auth_use_keystone = true
nss_db_path = {path to nss db}
```

Save the changes to the Ceph configuration file. Then, copy the updated Ceph configuration file to each Ceph node. For example:

```
# scp /etc/ceph/ceph.conf <node-name>:/etc/ceph/
```

See below for a detailed description of the available Keystone integration configuration options:

rgw_s3_auth_use_keystone

Description

If set to **true**, the Ceph Object Gateway will authenticate users using Keystone.

Type

Boolean

Default

false

nss_db_path

Description

The path to the NSS database.

Type

String

Default

""

rgw_keystone_url

Description

The URL for the administrative RESTful API on the Keystone server.

Type

String

Default

""

rgw_keystone_admin_token

Description

The token or shared secret that is configured internally in Keystone for administrative requests.

Type

String

Default

""

rgw_keystone_admin_user

Description

The keystone admin user name.

Type

String

Default

""

rgw_keystone_admin_password

Description

The keystone admin user password.

Type

String

Default

""

rgw_keystone_admin_tenant

Description

The Keystone admin user tenant for keystone v2.0.

Type

String

Default

""

rgw_keystone_admin_project

Description

The Keystone admin user project for keystone v3.

Type

String

Default

""

rgw_keystone_admin_domain

Description

The Keystone admin user domain.

Type

String

Default

""

rgw_keystone_api_version

Description

The version of the Keystone API to use. Valid options are **2** or **3**.

Type

Integer

Default

2

rgw_keystone_accepted_roles

Description

The roles required to serve requests.

Type

String

Default

"Member, admin"

rgw_keystone_accepted_admin_roles

Description

The list of roles allowing a user to gain administrative privileges.

Type

String

Default

""

rgw_keystone_token_cache_size

Description

The maximum number of entries in the Keystone token cache.

Type

Integer

Default

10000

rgw_keystone_revocation_interval

Description

The number seconds between tokens revocation check.

Type

Integer

Default

15 * 60

rgw_keystone_verify_ssl

Description
: If **true** Ceph will try to verify Keystone's SSL certificate.

Type
: Boolean

Default
: **true**

rgw_keystone_implicit_tenants

Description
: Create new users in their own tenants of the same name. Set this to **true** or **false** under most circumstances. For compatibility with previous versions of Red Hat Ceph Storage, it is also possible to set this to **s3** or **swift**. This has the effect of splitting the identity space such that only the indicated protocol will use implicit tenants. Some older versions of Red Hat Ceph Storage only supported implicit tenants with Swift.

Type
: String

Default
: **false**

2.3. RESTARTING CIVETWEB

Once you have saved the Ceph configuration file and distributed it to each Ceph node, restart the Ceph Object Gateway instances. Usage should be one of:

```
# systemctl restart ceph-radosgw
# systemctl restart ceph-radosgw@rgw.`hostname -s`
```