Keystone Installation for Ubuntu

Identity Service (Keystone) Overview

- The OpenStack Identity service provides a single point of integration for managing authentication, authorization, and a catalog of services.

- The OpenStack Identity service provides a single point of integration for managing authentication, authorization, and a catalog of services.

- The Identity service contains these components:

  Server: A centralized server provides authentication and authorization services using a RESTful interface.

  Drivers: Drivers or a service back end are integrated to the centralized server. They are used for accessing identity information in repositories external to OpenStack

  Modules: Middleware modules run in the address space of the OpenStack component that is using the Identity service. These modules intercept service requests, extract user credentials, and send them to the centralized server for authorization.

Prerequisites

Before you install and configure Keystone, you must create a database. But before you begin, ensure you have the most recent version of *python-pyasn1* installed.

1. Use the database access client to connect to the database server as the root user.

   mysql –u root –p

2. Create the *keystone* database.

   MariaDB [(none)]> CREATE DATABASE keystone;

3. Grant proper access to the *keystone* database.

   MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone. TO 'keystone'@'localhost' \ IDENTIFIED BY 'KEYSTONE_DBPASS';*

4. *Exit the database client.*

Install and Configure Components

1. Run the following command to install the packages.

   sudo zypper install openstack-keystone apache2-mod_wsgi

2. Edit the /etc/keystone/keystone.conf file and complete the following actions:
   - In the [database] section, configure database access:

     ```
     [database]
     # ...
     connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@controller/keystone
     ```

   - In the [token] section, configure the Fernet token provider:

     ```
     [token]
     # ...
     provider = fernet
     ```

3. Populate the Identity service database:

   su –s /bin/sh –c “keystone-manage db_sync” keystone

Install and Configure Components

4. Initialize the Fernet key repositories:

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

5. Bootstrap the Identity service:

```
keystone-manage bootstrap --bootstrap-password ADMIN_PASS \
--bootstrap-admin-url http://controller:35357/v3/ \
--bootstrap-internal-url http://controller:5000/v3/ \
--bootstrap-public-url http://controller:5000/v3/ \
--bootstrap-region-id RegionOne
```

Configure the Apache HTTP server

1. Edit the /etc/sysconfig/apache2 file and configure the APACHE_SERVERNAME option to reference the controller node:

   APACHE_SERVERNAME = “controller”

2. Create the /etc/apache2/conf.d/wsgi-keystone.conf file with the following content:

```
Listen 5000
Listen 35357
<VirtualHost *:5000>
    WSGIDaemonProcess keystone-public processes=5 threads=1
    user=keystone group=keystone display-name=%{GROUP}
    WSGIProcessGroup keystone-public
    WSGIScriptAlias / /usr/bin/keystone-wsgi-public
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
```

Configure the Apache HTTP Server

```
    ErrorLogFormat “%{cu}t %M”
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined
    <Directory /usr/bin>
        Require all granted
    </Directory>
</VirtualHost>
<VirtualHost *:35357>
    WSGIDaemonProcess keystone-admin processes=5 threads=1
    user=keystone group=keystone display-name=%{GROUP}
    WSGIProcessGroup keystone-admin
    WSGIScriptAlias / /usr/bin/keystone-wsgi-admin
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
```

Configure the Apache HTTP Server

```
    ErrorLogFormat “%{cu}t %M”
    ErrorLog /var/log/apache2/keystone.log
    CustomLog /var/log/apache2/keystone_access.log combined
    <Directory /usr/bin>
        Require all granted
    </Directory>
</VirtualHost>
```

3. Recursively change the ownership of the /etc/keystone directory:

```
chown –R keystone:keystone /etc/keystone
```

Finalize the Installation

1. Start the Apache HTTP service and configure it to start when the system boots:

   ```
   systemctl enable apache2.service
   systemctl start apache2.service
   ```

2. Configure the administrative account

   ```
   export OS_USERNAME = admin
   export OS_PASSWORD = ADMIN_PASS
   export OS_PROJECT_NAME=admin
   export OS_USER_DOMAIN_NAME=Default
   export OS_PROJECT_DOMAIN_NAME=Default
   export OS_AUTH_URL=http://controller:35357/v3
   export OS_IDENTITY_API_VERSION=3
   ```

Create a domain, project, users and roles

The Identity service provides authentication services for each OpenStack service. The authentication service uses a combination of domains, projects, users and roles.

1. Create a *service* project:

   openstack project create –domain default \ --description “Service Project” service

2. Create the *demo* project:

   openstack project create –domain default \ --description “Demo Project” demo

3. Create the demo user:

   openstack user create –domain default \ --password-prompt demo

4. Create the user role:

   openstack role create user

Create a domain, project, users and roles

5. Add the user role to the demo project and user:

   openstack role add –project demo –user demo user

Verifying Operation

1. For security reasons, disable the temporary authentication token mechanism:

   Edit the /etc/keystone/keystone-paste.ini file and remove admin_token_auth from the [pipeline: public_api], [public:admin_api], and [pipeline:api_v3] sections.

2. Unset the temporary OS_AUTH_URL and OS_PASSWORD environment variable:

   ```
   unset OS_AUTH_URL OS_PASSWORD
   ```

3. As the admin user, request an authentication token:

   ```
   openstack --os-auth-url http://controller:35357/v3 \
   --os-project-domain-name Default --os-user-domain-name Default \
   --os-project-name admin --os-username admin token issue
   ```

4. In the same way, request an authentication token as the demo user at port 5000.

Create OpenStack Client Environment Scripts

OpenStack supports simple client environment scripts aslo known as OpenRC files. These scripts contain common options for all clients, but also support unique options.

1. Create and edit the *admin-openrc* file and add the following content:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME = admin
export OS_PASSWORD = ADMIN_PASS
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

Create OpenStack Client Environment Scripts

2. Create and edit the *demo-openrc* file and add the following content:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME = demo
export OS_PASSWORD = DEMO_PASS
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

3. Using the scripts: Load the admin-openrc file to populate the environment variables with the location of the Identity service and the admin project and user credentials:

```
. admin-openrc
```

4. Request an authentication token:

```
openstack token issue
```