Red Hat OpenStack Platform 9
Users and Identity Management Guide

Managing users and authentication mechanisms

OpenStack Team

Red Hat OpenStack Platform 9 Users and Identity Management Guide

Managing users and authentication mechanisms

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

The Users and Identity Management Guide provides the procedures to manage user roles, quotas, projects and project security, and the Identity service of a Red Hat OpenStack Platform environment.

Table of Contents

PREFACE

As a cloud administrator, you can manage projects, users, and roles. Projects are organizational units in the cloud to which you can assign users. Projects are also known as tenants or accounts. Users can be members of one or more projects. Roles define the actions that users can perform.

Each OpenStack deployment must include at least one project, one user, and one role, linked together. As a cloud administrator, you can add, update, and delete projects and users, assign users to one or more projects, and change or remove these assignments. You can manage projects and users independently from each other.

You can also configure user authentication with the Keystone identity service to control access to services and endpoints. Keystone provides token-based authentication and can integrate with LDAP and Active Directory, so you can manage users and identities externally and synchronize the user data with Keystone.

CHAPTER 1. USER MANAGEMENT

1.1. USER MANAGEMENT

As a cloud administrator, you can add, modify, and delete users in the dashboard. Users can be members of one or more projects. You can manage projects and users independently from each other.

1.1.1. Create a User

Use this procedure to create users in the dashboard. You can assign a primary project and role to the user. Note that users created in the dashboard are Keystone users by default. To integrate Active Directory users, you can configure the LDAP provider included in the Red Hat OpenStack Platform Identity service.

1. As an admin user in the dashboard, select **Identity > Users**.
2. Click **Create User**.
3. Enter a user name, email, and preliminary password for the user.
4. Select a project from the **Primary Project** list.
5. Select a role for the user from the **Role** list (the default role is `_member_`).
6. Click **Create User**.

1.1.2. Edit a User

Use this procedure to update the user's details, including the primary project.

1. As an admin user in the dashboard, select **Identity > Users**.
2. In the User's **Actions** column, click **Edit**.
3. In the **Update User** window, you can update the **User Name**, **Email**, and **Primary Project**.
4. Click **Update User**.

1.1.3. Enable or Disable a User

Use this procedure to enable or disable a user. You can disable or enable only one user at a time. A disabled user cannot log in to the dashboard, and does not have access to any OpenStack services. Also, a disabled user's primary project cannot be set as active. A disabled user can be enabled again, unlike deleting a user where the action cannot be reversed. A disabled user must be re-enabled for any user-project action in the dashboard.

1. As an admin user in the dashboard, select **Identity > Users**.
2. In the **Actions** column, click the arrow, and select **Enable User** or **Disable User**. In the **Enabled** column, the value then updates to either `True` or `False`.

1.1.4. Delete a User

As an admin user, use this procedure to delete a user using the dashboard. This action cannot be reversed, unlike disabling a user. Deleted users get delisted from a project’s members' list for projects it belongs to. All roles associated with the user-project pair are also lost.

1. As an admin user in the dashboard, select **Identity > Users**.
2. Select the users you want to delete.
3. Click **Delete Users**. The **Confirm Delete Users** window is displayed.
4. Click **Delete Users** to confirm the action.

CHAPTER 2. ROLE MANAGEMENT

2.1. ROLE MANAGEMENT

OpenStack uses a role-based access control (RBAC) mechanism to manage access to its resources. Roles define which actions users can perform. By default, there are two predefined roles: a member role that gets attached to a tenant, and an administrative role to enable non-admin users to administer the environment. Note that there are abstract levels of permission, and it is possible to create the roles the administrator needs, and configure services adequately.

2.1.1. View Roles

Use the following command to list the available predefined roles.

```
$ keystone role-list
+----------------------------------+---------------+
|                id                |      name     |
+----------------------------------+---------------+
| 71ccc37d41c8491c975ae72676db687f |     Member    |
| 149f50a1fe684bfa88dae76a48d26ef7 | ResellerAdmin |
| 9fe2ff9ee4384b1894a90878d3e92bab |    _member_   |
| 6ecf391421604da985db2f141e46a7c8 |     admin     |
+----------------------------------+---------------+
```

To get details for a specified role, run:

```
$ keystone role-get [ROLE]
```

Example

```
$ keystone role-get admin
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 6ecf391421604da985db2f141e46a7c8 |
|   name   |             admin                |
+----------+----------------------------------+
```

2.1.2. Create and Assign a Role

As a cloud administrator, you can create and manage roles on the Keystone client using the following set of commands. Each OpenStack deployment must include at least one project, one user, and one role, linked together. However, users can be members of multiple projects. To assign users to multiple projects, create a role and assign that role to a user-project pair. Note that you can create a user and assign a primary project and default role in the dashboard.

Note

Either the name or ID can be used to specify users, roles, or projects.

1. Create the **new-role** role:

```
$ keystone role-create --name [ROLE_NAME]
```

Example

```
$ keystone role-create --name new-role
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | 61013e7aa4ba4e00a0a1ab4b14bc6b2a |
|   name   |             new-role             |
+----------+----------------------------------+
```

2. To assign a user to a project, you must assign the role to a user-project pair. To do this, obtain the user, role, and project names or IDs:

   a. List users:

   ```
   $ keystone user-list
   ```

   b. List roles:

   ```
   $ keystone role-list
   ```

   c. List projects:

   ```
   $ keystone tenant-list
   ```

3. Assign a role to a user-project pair.

```
$ keystone user-role-add --user [USER_NAME] --role [ROLE_NAME] --
tenant [TENANT_NAME]
```

Example

In this example, you assign the **new-role** role to the **demo**-**demo** pair:

```
$ keystone user-role-add --user demo --role new-role --tenant
demo
```

4. Verify the role assignment for the user **demo**:

```
$ keystone user-role-list --user [USER_NAME] --tenant
[TENANT_NAME]
```

Example

```
$ keystone user-role-list --user demo --tenant demo
```

2.1.3. Delete a Role

1. Use the following command to delete a role from a user-project pair. Deleting a role ensures the associated user-project pairing is lost.

```
$ keystone user-role-remove --user [USER_NAME] --role [ROLE] --
tenant [TENANT_NAME]
```

2. Verify the role removal:

```
$ keystone user-role-list --user [USER_NAME] --tenant
[TENANT_NAME]
```

   If the role was removed, the command output omits the removed role.

CHAPTER 3. GROUP MANAGEMENT

3.1. MANAGE KEYSTONE GROUPS

3.1.1. Using the Command-line

You can use Identity Service (keystone) groups to assign consistent permissions to multiple user accounts. This example creates a group and then assigns permissions to the group. As a result, members of the group will inherit the same permissions that were assigned to the group:

Note

The **`openstack group`** subcommands require keystone **`v3`**.

1. Create the group **`grp-Auditors`**:

```
$ openstack group create grp-Auditors
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | 2a4856fc242142a4aa7c02d28edfdfff |
| name        | grp-Auditors                     |
+-------------+----------------------------------+
```

2. View a list of keystone groups:

```
$ openstack group list --long
+----------------------------------+-------------+-----------
+-------------+
| ID                               | Name         | Domain ID |
Description |
+----------------------------------+-------------+-----------
+-------------+
| 2a4856fc242142a4aa7c02d28edfdfff | grp-Auditors | default   |
|
+----------------------------------+-------------+-----------
+-------------+
```

3. Grant the **`grp-Auditors`** group permission to access the **`demo`** project, while using the **`_member_`** role:

```
$ openstack role add _member_ --group grp-Auditors --project demo
```

4. Add the existing user **`user1`** to the **`grp-Auditors`** group:

```
$ openstack group add user grp-Auditors user1
user1 added to group grp-Auditors
```

5. Confirm that **`user1`** is a member of **`grp-Auditors`**:

```
$ openstack group contains user grp-Auditors user1
user1 in group grp-Auditors
```

6. Review the effective permissions that have been assigned to **`user1`**:

```
$ openstack role assignment list --effective --user user1
+----------------------------------+------------------------
---------+-------+----------------------------------+--------
+-----------+
| Role                             | User
| Group | Project                          | Domain | Inherited
|
+----------------------------------+------------------------
---------+-------+----------------------------------+--------
+-----------+
| 9fe2ff9ee4384b1894a90878d3e92bab |
3fefe5b4f6c948e6959d1feaef4822f2 |       |
0ce36252e2fb4ea8983bed2a568fa832 |        | False     |
+----------------------------------+------------------------
---------+-------+----------------------------------+--------
+-----------+
```

3.1.2. Using Dashboard

You can use the dashboard to manage the membership of keystone groups. You will need to use the command-line to assign role permissions to a group, as covered in the previous example.

3.1.2.1. Create a Group

1. As an admin user in the dashboard, select **Identity > Groups**.
2. Click **+Create Group**.
3. Enter a name and description for the group.
4. Click **Create Group**.

3.1.2.2. Manage Group Membership

You can use the dashboard to manage the membership of keystone groups.

1. As an admin user in the dashboard, select **Identity > Groups**.
2. Click **Manage Members** for the group you need to edit.
3. Use **Add users** to add a user to the group. If you need to remove a user, mark its checkbox and click or **Remove users**.

CHAPTER 4. QUOTA MANAGEMENT

4.1. QUOTA MANAGEMENT

As a cloud administrator, you can set and manage quotas for a project. Each project is allocated resources, and project users are granted access to consume these resources. This enables multiple projects to use a single cloud without interfering with each other's permissions and resources. A set of resource quotas are preconfigured when a new tenant is created. The quotas include the amount of VCPUs, instances, RAM, floating IPs, that can be assigned to tenants. Quotas can be enforced at both the tenant (or project) and the tenant-user level. Note that you can set or modify Compute and Block Storage quotas for new and existing tenants using the dashboard. See Chapter 5, *Project Management* for the procedure on how to set and update project quotas within the dashboard.

4.1.1. View Compute Quotas for a User

Run the following command to list the currently set quota values for a user:

```
$ nova quota-show --user [USER] --tenant [TENANT]
```

Example

```
$ nova quota-show --user demoUser --tenant demo
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 5     |
| fixed_ips                   | -1    |
| metadata_items              | 128   |
| injected_files              | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes    | 255   |
| key_pairs                   | 100   |
| security_groups             | 10    |
| security_group_rules        | 20    |
| server_groups               | 10    |
| server_group_members        | 10    |
+-----------------------------+-------+
```

4.1.2. Update Compute Quotas for a User

Run the following commands to update a particular quota value:

```
$ nova quota-update --user [USER] --[QUOTA_NAME] [QUOTA_VALUE] [TENANT]
$ nova quota-show --user [USER] --tenant [TENANT]
```

Example

```
$ nova quota-update --user demoUser --floating-ips 10 demo
$ nova quota-show --user demoUser --tenant demo
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 10    |
| ...                         |       |
+-----------------------------+-------+
```

Note

To view a list of options for the quota-update command, run:

```
$ nova help quota-update
```

4.1.3. Set Object Storage Quotas for a User

Object Storage quotas can be classified under the following categories:

- Container quotas - Limits the total size (in bytes) or number of objects that can be stored in a single container.
- Account quotas - Limits the total size (in bytes) that a user has available in the Object Storage service.

To set either container quotas or the account quotas, the Object Storage proxy server must have the parameters **`container_quotas`** or **`account_quotas`** (or both) added to the **`[pipeline:main]`** section of the **`proxy-server.conf`** file:

```
[pipeline:main]
pipeline = catch_errors [...] tempauth container-quotas \
account-quotas slo dlo proxy-logging proxy-server

[filter:account_quotas]
use = egg:swift#account_quotas

[filter:container_quotas]
use = egg:swift#container_quotas
```

Use the following command to view and update the Object Storage quotas. All users included in a project can view the quotas placed on the project. To update the Object Storage quotas on a project, you must have the role of a ResellerAdmin in the project.

To view account quotas:

```
# swift stat

Account: AUTH_b36ed2d326034beba0a9dd1fb19b70f9
Containers: 0
Objects: 0
```

```
Bytes: 0
Meta Quota-Bytes: 214748364800
X-Timestamp: 1351050521.29419
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

To update quotas:

```
# swift post -m quota-bytes:<BYTES>
```

For example, to place a 5 GB quota on an account:

```
# swift post -m quota-bytes:5368709120
```

To verify the quota, run the **`swift stat`** command again:

```
# swift stat

Account: AUTH_b36ed2d326034beba0a9dd1fb19b70f9
Containers: 0
Objects: 0
Bytes: 0
Meta Quota-Bytes: 5368709120
X-Timestamp: 1351541410.38328
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

CHAPTER 5. PROJECT MANAGEMENT

5.1. PROJECT MANAGEMENT

As a cloud administrator, you can create and manage projects (tenants). A tenant describes a project with an assigned number of OpenStack users and resources. It is possible to set up quotas for each tenant. This enables multiple projects to use a single cloud without interfering with each other's permissions and resources. The words project and tenant are used interchangeably. Users can be associated with more than one project. Each user-project pairing must have a role associated with it.

5.1.1. Create a Project

Use this procedure to create projects, add members to the project, and set resource limits for the project.

1. As an admin user in the dashboard, select **Identity > Projects**.
2. Click **Create Project**.
3. On the **Project Information** tab, enter a name and description for the project (the **Enabled** check box is selected by default).
4. On the **Project Members** tab, add members to the project from the **All Users** list.
5. On the **Quotas** tab, specify resource limits for the project.
6. Click **Create Project**.

5.1.2. Edit a Project

You can edit a project to change its name or description, enable or temporarily disable it, or update its members.

1. As an admin user in the dashboard, select **Identity > Projects**.
2. In the project's **Actions** column, click the arrow, and click **Edit Project**.
3. In the **Edit Project** window, you can update a project to change its name or description, and enable or temporarily disable the project.
4. On the **Project Members** tab, add members to the project, or remove them as needed.
5. Click **Save**.

Note

The **Enabled** check box is selected by default. To temporarily disable the project, clear the **Enabled** check box. To enable a disabled project, select the **Enabled** check box.

5.1.3. Delete a Project

1. As an admin user in the dashboard, select **Identity > Projects**.
2. Select the project you want to delete.
3. Click **Delete Projects**. The **Confirm Delete Projects** window is displayed.
4. Click **Delete Projects** to confirm the action.

The project gets deleted and any user pairing will be disassociated.

5.1.4. Update Project Quotas

Quotas are operational limits that can be set per project to optimize cloud resources. You can set quotas to prevent project resources from being exhausted without notification. Quotas can be enforced at both the project and the project-user level.

1. As an admin user in the dashboard, select **Identity > Projects**.
2. In the project's **Actions** column, click the arrow, and click **Modify Quotas**.
3. In the **Quota** tab, modify project quotas as needed.
4. Click **Save**.

5.1.5. Change Active Project

A user can set a project as the active project only of which they are a member. It is also necessary for the user to be a member of more than one project to have the **Set as Active Project** option be enabled. Setting a project as an active project enables you to access objects in the dashboard for the active project. Note that a disabled project cannot be set as active, unless it is re-enabled.

1. As an admin user in the dashboard, select **Identity > Projects**.
2. In the project's **Actions** column, click the arrow, and click **Set as Active Project**.
3. Alternatively, as a non-admin user, in the project's **Actions** column, click **Set as Active Project** which becomes the default action in the column.

5.2. PROJECT SECURITY MANAGEMENT

Security groups are sets of IP filter rules that can be assigned to project instances, and which define networking access to the instance. Security groups are project specific; project members can edit the default rules for their security group and add new rule sets.

All projects have a default security group that is applied to any instance that has no other defined security group. Unless you change the default values, this security group denies all incoming traffic and allows only outgoing traffic to your instance.

5.2.1. Create a Security Group

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, click **Create Security Group**.

3. Provide a name and description for the group, and click **Create Security Group**.

5.2.2. Add a Security Group Rule

By default, rules for a new group only provide outgoing access. You must add new rules to provide additional access.

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, click **Manage Rules** for the security group that you want to edit.
3. Click **Add Rule** to add a new rule.
4. Specify the rule values, and click **Add**.

   The following rule fields are required:

   Rule

   Rule type. If you specify a rule template (for example, *SSH*), its fields are automatically filled in:

   - TCP: Typically used to exchange data between systems, and for end-user communication.
   - UDP: Typically used to exchange data between systems, particularly at the application level.
   - ICMP: Typically used by network devices, such as routers, to send error or monitoring messages.

   Direction

   Ingress (inbound) or Egress (outbound).

   Open Port

   For TCP or UDP rules, the **Port** or **Port Range** (single port or range of ports) to open:

   - For a range of ports, enter port values in the **From Port** and **To Port** fields.
   - For a single port, enter the port value in the **Port** field.

   Type

   The type for ICMP rules; must be in the range *-1:255*.

   Code

   The code for ICMP rules; must be in the range *-1:255*.

   Remote

   The traffic source for this rule:

   - CIDR (Classless Inter-Domain Routing): IP address block, which limits access to IPs within the block. Enter the CIDR in the Source field.

- Security Group: Source group that enables any instance in the group to access any other group instance.

5.2.3. Delete a Security Group Rule

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, click **Manage Rules** for the security group.
3. Select the security group rule, and click **Delete Rule**.
4. Click **Delete Rule** again.

Note

You cannot undo the delete action.

5.2.4. Delete a Security Group

1. In the dashboard, select **Project > Compute > Access & Security**.
2. On the **Security Groups** tab, select the group, and click **Delete Security Groups**.
3. Click **Delete Security Groups**.

Note

You cannot undo the delete action.

5.3. HIERARCHICAL MULTI-TENANCY IN IDENTITY SERVICE

Multi-tenancy is an architecture in which a single instance of a software application serves multiple customers. In cloud computing, the meaning of multi-tenancy architecture has broadened because of new service models that take advantage of virtualization and remote access. A software-as-a-service (SaaS) provider, for example, can run one instance of its application on one instance of a database and provide web access to multiple customers. In such a scenario, each tenant's data is isolated and remains invisible to other tenants.

In the OpenStack Identity Service (**`keystone`**) you can use multi-tenancy to nest projects. Domains represent collections of users, groups, and projects where each one of these is owned by exactly one domain. Users can be associated with multiple projects by granting roles to them on a project, including projects owned by other domains. Projects are the container of resources, which define quotas and access to VM images.

Note

Multi-tenancy is available as a technology preview. For more information on the support scope for features marked as technology previews, see https://access.redhat.com/support/offerings/techpreview/

CHAPTER 6. DOMAIN MANAGEMENT

Identity Service (keystone) domains are additional namespaces you can create in keystone. You would use keystone domains to partition users, groups, and projects. These separate domains can also be configured to authenticate users in different LDAP or Active Directory environments. For more information see Integrate with Identity Service.

Note

Identity Service includes a built-in domain called **`Default`**. It is suggested you reserve this domain only for service accounts, and create a separate domain for user accounts.

6.1. VIEW A LIST OF DOMAINS

You can view a list of domains using **`openstack domain list`**. For example:

```
$ openstack domain list
+----------------------------------+------------------+---------+--
------------------+
| ID                               | Name             | Enabled |
Description        |
+----------------------------------+------------------+---------+--
------------------+
| 3abefa6f32c14db9a9703bf5ce6863e1 | TestDomain       | True    |
|
| 69436408fdcb44ab9e111691f8e9216d | corp             | True    |
|
| a4f61a8feb8d4253b260054c6aa41adb | federated_domain | True    |
|
| default                          | Default          | True    | The
default domain |
+----------------------------------+------------------+---------+--
------------------+
```

Note

If this command is not available, check you have enabled keystone v3 for your command line session.

6.2. CREATE A NEW DOMAIN

You can create a new domain using **`openstack domain create`**. For example:

```
$ openstack domain create TestDomain
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
```

```
| enabled     | True                             |
| id          | 3abefa6f32c14db9a9703bf5ce6863e1 |
| name        | TestDomain                       |
+-------------+----------------------------------+
```

6.3. VIEW THE DETAILS OF A DOMAIN

You can view the details of a domain using **`openstack domain show`**. For example:

```
$ openstack domain show TestDomain
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 3abefa6f32c14db9a9703bf5ce6863e1 |
| name        | TestDomain                       |
+-------------+----------------------------------+
```

6.4. DISABLE A DOMAIN

1. You can disable a domain using **`--disable`**. For example:

```
$ openstack domain set TestDomain --disable
```

2. Confirm the domain has been disabled:

```
$ openstack domain show TestDomain
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | False                            |
| id          | 3abefa6f32c14db9a9703bf5ce6863e1 |
| name        | TestDomain                       |
+-------------+----------------------------------+
```

3. You can then re-enable the domain, if required:

```
$ openstack domain set TestDomain --enable
```

CHAPTER 7. IDENTITY MANAGEMENT

7.1. SECURE LDAP COMMUNICATION

If you have configured the Identity service (keystone) to authenticate against or to retrieve identity information from an LDAP server, you can secure LDAP communication for the Identity service using a CA certificate.

This section outlines how to obtain the CA certificate from Active Directory, how to convert the CA certificate file into Privacy Enhanced Mail (PEM) file format, and the three methods for configuring secure LDAP communication for the Identity service. The procedure in each method must be performed depending on where and how the CA trust is configured.

7.1.1. Obtaining the CA Certificate from Active Directory

The following code shows an example of how to query Active Directory to obtain the CA certificate. The CA_NAME is the name of the certificate (you can see it in mmc.exe) and the rest of the parameters can be changed according to your setup:

```
CA_NAME="WIN2012DOM-WIN2012-CA"
AD_SUFFIX="dc=win2012dom,dc=com"
LDAPURL="ldap://win2012.win2012dom.com"
ADMIN_DN="cn=Administrator,cn=Users,$AD_SUFFIX"
ADMINPASSWORD="MyPassword"

CA_CERT_DN="cn=latexmath:[$CA_NAME,cn=certification
authorities,cn=public key
services,cn=services,cn=configuration,$]AD_SUFFIX"

TMP_CACERT=/tmp/cacert.`date +'%Y%m%d%H%M%S'`.$$.pem

ldapsearch -xLLL -H
latexmath:[$LDAPURL -D `echo \"$]ADMIN_DN"`-W -s base -b`echo
"$CA_CERT_DN"` objectclass=* cACertificate
```

7.1.2. Converting the CA Certificate into PEM file format

Create a file called /path/cacert.pem and include the contents of the LDAP query — that obtained the CA certificate from Active Directory, within the header and footer, as shown in the example below:

```
-----BEGIN CERTIFICATE-----
MIIDbzCCAlegAwIBAgIQQD14hh1Yz7tPFLXCkKUOszANB... -----END
CERTIFICATE-----
```

For troubleshooting, you can execute the following query to check if LDAP is working, and to ensure the PEM certificate file was created correctly.

```
LDAPTLS_CACERT=/path/cacert.pem ldapsearch -xLLL -ZZ -H $LDAPURL -s
base -b "" "objectclass=*" currenttime
```

The query should return a result similar to:

```
dn: currentTime:
20141022050611.0Z
```

You can run the following command to get a CA certificate if it was hosted by a web server.

Example

- $HOST=redhat.com
- $PORT=443

```
# echo Q | openssl s_client -connect $HOST:$PORT | sed -n -e
'/BEGIN CERTIFICATE/,/END CERTIFICATE/ p'
```

7.1.3. Methods for Configuring Secure LDAP Communication for the Identity Service

7.1.3.1. Method 1

Use this method if the CA trust is configured at the LDAP level using a PEM file. Manually specify the location of a CA certificate file. The following procedure secures LDAP communication not only for the Identity service, but for all applications that use the OpenLDAP libraries.

1. Copy the file containing your CA certificate chain in PEM format to the **`/etc/openldap/certs`** directory.
2. Edit **`/etc/openldap/ldap.conf`** and add the following directive, replacing [CA_FILE] with the location and name of the CA certificate file:

   ```
   TLS_CACERT /etc/openldap/certs/[CA_FILE]
   ```

3. Restart the **openstack-keystone** service:

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

7.1.3.2. Method 2

Use this method if the CA trust is configured at the LDAP library level using a Network Security Services (NSS) database. Use the **`certutil`** command to import and trust a CA certificate into the NSS certificate database used by the OpenLDAP libraries. The following procedure secures LDAP communication not only for the Identity service, but for all applications that use the OpenLDAP libraries.

1. Import and trust the certificate, replacing [CA_FILE] with the location and name of the CA certificate file:

   ```
   # certutil -d /etc/openldap/certs -A -n "My CA" -t CT,, -a -i
   [CA_FILE]
   ```

2. Confirm the CA certificate was imported correctly:

```
# certutil -d /etc/openldap/certs -L
```

Your CA certificate is listed, and the trust attributes are set to **CT,,**.

3. Restart the **openstack-keystone** service:

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

7.1.3.3. Method 3

Use this method if the CA trust is configured at the Keystone level using a PEM file. The final method of securing communication between the Identity service and an LDAP server is to configure TLS for the Identity service.

However, unlike the two methods above, this method only secures LDAP communication for the Identity service and does not secure LDAP communication for other applications that use the OpenLDAP libraries.

The following procedure uses the **`openstack-config`** command to edit values in the **`/etc/keystone/keystone.conf`** file.

1. Enable TLS:

```
# openstack-config --set /etc/keystone/keystone.conf ldap use_tls
True
```

2. Specify the location of the certificate, replacing [CA_FILE] with the name of the CA certificate:

```
# openstack-config --set /etc/keystone/keystone.conf ldap
tls_cacertfile [CA_FILE]
```

3. Specify the client certificate checks performed on incoming TLS sessions from the LDAP server, replacing [CERT_BEHAVIOR] with one of the behaviors listed below:

demand

a certificate will always be requested from the LDAP server. The session will be terminated if no certificate is provided, or if the certificate provided cannot be verified against the existing certificate authorities file.

allow

a certificate will always be requested from the LDAP server. The session will proceed as normal even if a certificate is not provided. If a certificate is provided but it cannot be verified against the existing certificate authorities file, the certificate will be ignored and the session will proceed as normal.

never

a certificate will never be requested.

```
# openstack-config --set /etc/keystone/keystone.conf ldap
tls_req_cert [CERT_BEHAVIOR]
```

4. Restart the **openstack-keystone** service:

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