Red Hat OpenStack Platform 16.1

用户和身份管理指南

管理用户和 keystone 身份验证

Last Updated: 2023-09-19

Red Hat OpenStack Platform 16.1 用户和身份管理指南

管理用户和 keystone 身份验证

OpenStack Team
rhos-docs@redhat.com

法律通告

摘要

管理应用凭据、用户、角色、项目和配额。

目录

前言

注意

您不能在实例创建过程中将基于角色的访问控制(RBAC)共享安全组直接应用到实例。要将 RBAC 共享安全组应用到实例，必须首先创建端口，将共享安全组应用到该端口，然后将该端口分配给实例。请参阅 将安全组添加到端口。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 信息 . :leveloffset: +0

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

使用直接文档反馈(DDF)功能

使用 **添加反馈** DDF 功能，用于特定句子、段落或代码块上的直接注释。

1. 以 *Multi-page HTML* 格式查看文档。
2. 请确定您看到文档右上角的 **反馈** 按钮。
3. 用鼠标指针高亮显示您想评论的文本部分。
4. 点 **添加反馈**。
5. 在**添加反馈**项中输入您的意见。
6. 可选：添加您的电子邮件地址，以便文档团队可以联系您以讨论您的问题。
7. 点 **Submit**。

第 1 章 前言

Identity 服务

作为云管理员，您可以管理项目、用户和角色。

项目是包含一组资源的组织单元。您可以将用户分配给项目中的角色。角色定义那些用户可以对给定项目中的资源执行的操作。用户可以在多个项目中分配角色。

每个 Red Hat OpenStack (RHOSP)部署必须至少包含一个分配给项目中的角色的用户。作为云管理员，您可以：

- 添加、更新和删除项目和用户。
- 将用户分配给一个或多个角色，并更改或删除这些分配。
- 相互独立管理项目和用户。

您还可以使用 Identity 服务(keystone)配置用户身份验证，以控制对服务和端点的访问。Identity 服务提供基于令牌的身份验证，并可与 LDAP 和 Active Directory 集成，以便您可以外部管理用户和身份，并将用户数据与 Identity 服务同步。

第 2 章 管理用户

作为云管理员，您可以在仪表板中添加、修改和删除用户。用户可以是一个或多个项目的成员。您可以独立管理项目和用户。

2.1. 使用仪表板创建用户

您可以为用户分配一个主项目和角色。使用 OpenStack Dashboard (horizon)创建的用户默认为 Identity 服务用户。您可以通过配置 Identity 服务中包含的 LDAP 供应商来集成 Active Directory 用户。

流程

1. 以 admin 用户身份登录到控制面板。
2. 选择 **Identity > Users**。
3. **单击创建用户**。
4. 输入用户的用户名、电子邮件和初始密码。
5. 从 **Primary Project** 列表中选择一个项目。
6. 从 **Role** 列表中选择用户的角色。默认角色是 **member**。
7. **单击创建用户**。

2.2. 使用仪表板编辑用户

您可以更新用户详情，包括主项目。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Identity > Users**。
3. 在 **Actions** 列中，点 **Edit**。
4. 在 **Update User** 窗口中，您可以更新 **User Name**, **Email**, and **Primary Project**。
5. 单击 **Update User**。

2.3. 使用仪表板启用或禁用用户

您可以使用仪表板禁用用户。此操作不可逆，这与删除用户不同。

限制：

- 您不能一次禁用或启用多个用户。
- 您不能将用户的主项目设置为 active。

其结果是，您禁用的用户无法禁用：

- 登录控制面板。

- 获取 RHOSP 服务的访问权限。
- 在仪表板中执行任何 user-project 操作。

流程

1. 在仪表板中作为 admin 用户，选择 **Identity > Users**。
2. 在 **Actions** 列中，单击箭头，然后选择 **Enable User** 或 **Disable User**。在 **Enabled** 列中，该值会更新为 **True** 或 **False**。

2.4. 使用仪表板删除用户

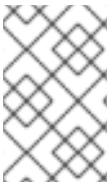
您必须是具有管理角色的用户的用户才能删除其他用户。无法撤销此操作。

1. 在仪表板中作为 admin 用户，选择 **Identity > Users**。
2. 选择您要删除的用户。
3. 单击 **Delete Users**。此时会显示 **Confirm Delete Users** 窗口。
4. 单击 **Delete Users** 以确认操作。

第 3 章 管理角色

Red Hat OpenStack Platform (RHOSP)使用基于角色的访问控制(RBAC)机制来管理对其资源的访问。Role 定义用户可以执行的操作。默认情况下，有两个预定义的角色：

- 附加到项目的 member 角色。
- 用于管理环境的管理员角色，使非管理员用户能够管理环境。



注意

Identity service (keystone)也添加了 **reader** 角色，该角色将显示在角色列表中。不要使用 **reader** 角色，因为它没有集成到其他 OpenStack 项目中，并在服务之间提供不一致的权限。

您还可以创建特定于环境的自定义角色。

3.1. 了解 RED HAT OPENSTACK PLATFORM 管理员角色

当您为用户授予 **admin** 角色时，此用户具有查看、更改、创建或删除任何项目的任何资源的权限。此用户可以创建可在项目间访问的共享资源，如公开可用的 glance 镜像或提供商网络。此外，具有 **admin** 角色的用户也可以创建和删除用户并管理角色。

您为其分配用户 **admin** 角色的项目是执行 **openstack** 命令的默认项目。例如，如果名为 **development** 的项目中的 **admin** 用户运行以下命令，则会在 **development** 项目中创建一个名为 **internal-network** 的网络：

```
openstack network create internal-network
```

admin 用户可以使用 **--project** 参数在任何项目中创建 **internal-network** ：

```
openstack network create internal-network --project testing
```

3.2. 使用 CLI 查看角色

作为管理员，您可以查看现有角色的详情

流程

1. 列出可用的预定义角色：

```
$ openstack role list
+----------------------------------+----------------+
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator  |
| ef3d3f510a474d6c860b4098ad658a29 | service        |
+----------------------------------+----------------+
```

2. 查看指定角色的详情：

```
$ openstack role show admin
```

Example

```
$ openstack role show admin
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | None                             |
| id        | 01d92614cd224a589bdf3b171afc5488 |
| name      | admin                            |
+-----------+----------------------------------+
```

注意

要获取与每个角色关联的权限的详细信息，您必须审核其对每个 API 调用的访问权限。如需更多信息，请参阅 审计 API 访问。

3.3. 使用 CLI 创建并分配角色

作为管理员，您可以通过以下一组命令，使用 Identity 服务(keystone)客户端创建和管理角色：每个 Red Hat OpenStack Platform 部署必须至少包含一个项目、一个用户和一个角色，链接在一起。

您可以将用户分配给多个项目。要将用户分配给多个项目，请创建一个角色，并将该角色分配给用户定义的项目对。

注意

您可以使用名称或 ID 来指定用户、角色或项目。

流程

1. 创建 **new-role** 角色：

```
$ openstack role create <role_name>
```

2. 要为项目分配用户，首先使用以下命令查找用户、角色和项目名称或 ID：
   - OpenStack 用户列表
   - OpenStack 角色列表
   - OpenStack 项目列表
3. 为用户分配角色分配给用户项目对。

```
$ openstack role add <role_name> --user <user_name> --project <project_name>
```

以下示例将 **admin** 角色分配给 **demo** 项目中的 **admin** 用户：

```
$ openstack role add admin --user admin --project demo
```

4. 验证用户 **admin** 的角色分配：

```
$ openstack role assignment list --user <user_name> --project <project_name> --names
```

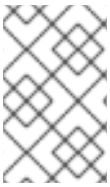
以下示例验证 **admin** 用户是否已分配给带有 **admin** 角色的 **demo** 项目。

```
$ openstack role assignment list --user admin --project demo --names
+-------+--------------+-------+-------------+--------+--------+-----------+
| Role  | User         | Group | Project     | Domain | System | Inherited |
+-------+--------------+-------+-------------+--------+--------+-----------+
| admin | admin@Default |      | demo@Default |        |        | False     |
+-------+--------------+-------+-------------+--------+--------+-----------+
```

3.4. 表示的角色

Identity 服务(keystone)强制访问控制，确认用户是否已分配给特定角色。Identity 服务使用 implied 角色分配。如果您明确将用户分配给角色，也可以隐式将用户分配给其他角色。您可以在 Red Hat OpenStack Platform 中查看默认的含义角色：

```
$ openstack implied role list
+----------------------------------+----------------+----------------------------------+------------------+
| Prior Role ID                    | Prior Role Name | Implied Role ID                  | Implied Role Name |
+----------------------------------+----------------+----------------------------------+------------------+
| 54454217f38247e5a2131c8a47138d32 | admin          | b59703369e194123b5c77dad60d11a25 | member           |
| b59703369e194123b5c77dad60d11a25 | member         | 382761de4a9c4414b6f8950f8580897c | reader           |
+----------------------------------+----------------+----------------------------------+------------------+
```



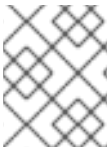
注意

Identity service (keystone)也添加了 **reader** 角色，该角色将显示在角色列表中。不要使用 **reader** 角色，因为它没有集成到其他 OpenStack 服务中，并在服务间提供不一致的权限。

具有更高权限的角色，与具有较少权限的角色相关联。在上面的默认表示的角色中，admin 表示 member，member 表示 reader。使用表示的角色时，用户的角色分配会被累积处理，以便用户继承下级角色。

3.4.1. 创建含义的角色

如果使用自定义角色，您可以创建指定的关联。



注意

当您创建新角色时，它将默认具有与 **member** 角色相同的访问策略。有关为自定义角色创建唯一策略的详情，请参考 使用策略文件进行访问控制。

流程

- 使用以下命令指定代表另一个角色的角色：

```
$ openstack implied role create manager --implied-role poweruser
+------------+----------------------------------+
| Field      | Value                            |
+------------+----------------------------------+
| implies    | ab0b966e0e5e411f8d8b0cc6c26fefd1 |
| prior_role | 880761f64bff4e4a8923efda73923b7a |
+------------+----------------------------------+
```

验证

- 列出所有含义的角色：

```
$ openstack implied role list
+----------------------------------+----------------+----------------------------------+------------------+
| Prior Role ID                    | Prior Role Name | Implied Role ID                  | Implied Role Name
|
+----------------------------------+----------------+----------------------------------+------------------+
| 54454217f38247e5a2131c8a47138d32 | admin          |
b59703369e194123b5c77dad60d11a25 | member           |
| 880761f64bff4e4a8923efda73923b7a | manager        |
ab0b966e0e5e411f8d8b0cc6c26fefd1 | poweruser        |
| b59703369e194123b5c77dad60d11a25 | member         |
382761de4a9c4414b6f8950f8580897c | reader           |
+----------------------------------+----------------+----------------------------------+------------------+
```

如果出错中进行了指示的关联，您可以撤销您的更改：

```
openstack implied role delete manager --implied-role poweruser
```

第 4 章 管理组

您可以使用 Identity Service (keystone)组为多个用户帐户分配一致的权限。

4.1. 使用命令行

创建组，并为组分配权限。组成员继承您分配给组中的相同权限：

1. 创建组 **grp-Auditors** ：

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

2. 查看 keystone 组列表：

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

3. 在使用 **member** 角色时，授予 **grp-Auditors** 组权限来访问 **demo** 项目：

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

4. 将现有用户 **user1** 添加到 **grp-Auditors** 组中：

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

5. 确认 **user1** 是 **grp-Auditors** 的成员：

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

6. 查看分配给 **user1** 的有效权限：

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

```
0ce36252e2fb4ea8983bed2a568fa832 |        | False    |
+----------------------------------+----------------------------------+-------+-------------------------------
--+--------+-----------+
```

4.2. 使用仪表板

您可以使用控制面板来管理 keystone 组成员。但是，您必须使用命令行为组分配角色权限。如需更多信息，请参阅 使用命令行。

4.2.1. 创建组

1. 以具有管理特权的用户身份登录控制面板。
2. 选择 **Identity > Groups**。
3. 单击 **+Create Group**。
4. 输入组的名称和描述。
5. 单击 **Create Group**。

4.2.2. 管理组成员资格

您可以使用控制面板来管理 keystone 组成员。

1. 以具有管理特权的用户身份登录控制面板。
2. 选择 **Identity > Groups**。
3. 点您要编辑的组的 **Manage Members**。
4. 使用 **Add users** 将用户添加到组中。如果要删除用户，请标记其复选框并点 **Remove users**。

第 5 章 配额管理

作为云管理员，您可以为项目设置和管理配额。每个项目分配资源，项目用户被授予使用这些资源的访问权限。这可让多个项目使用单个云，而不会相互干扰的权限和资源。创建新项目时，会预先配置一组资源配额。配额包括可分配给项目的过期、实例、RAM 和浮动 IP 数量。配额可以在项目和项目级别强制实施。您可以使用控制面板为新项目和现有项目设置或修改 Compute 和 Block Storage 配额。如需更多信息，请参阅 第 6 章 *项目管理*。

5.1. 查看用户的计算配额

运行以下命令列出用户当前设置的配额值：

```
$ 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    |
+-----------------------------+-------+
```

5.2. 更新用户的计算配额

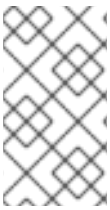
运行以下命令以更新特定的配额值：

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

示例

```
$ 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    |
| ...                         |       |
+-----------------------------+-------+
```



注意

要查看 quota-update 命令的选项列表，请运行：

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

5.3. 为用户设置对象存储配额

对象存储配额可分为以下类别：

- 容器配额 - 限制单个容器中可以存储的总大小（以字节为单位）或对象数量。
- 帐户配额 - 限制用户在对象存储服务中可用的总大小（以字节为单位）。

要设置容器配额或帐户配额，对象存储代理服务器必须具有添加到 **proxy-server.conf** 文件的 **[pipeline:main]** 部分的参数 **container_quotas** 或 **account_quotas**（或两者）：

```
[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
```

使用以下命令查看和更新对象存储配额。项目中包含的所有用户都可以查看项目中放置的配额。要更新项目中的 Object Storage 配额，您必须在项目中具有 ResellerAdmin 角色。

查看帐户配额：

```
# 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
```

更新配额：

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

例如，将 5 GB 配额放在帐户中：

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

第 6 章 项目管理

6.1. 项目管理

作为云管理员，您可以创建和管理项目。项目是共享虚拟资源池，您可以为它分配 OpenStack 用户和组。您可以在每个项目中配置共享虚拟资源的配额。您可以使用 Red Hat OpenStack Platform 创建多个项目，它不会相互影响的权限和资源。用户可以与多个项目关联。每个用户都必须为为其分配的每个项目分配一个角色。

6.1.1. 创建一个项目

创建一个项目，添加成员到项目，并为项目设置资源限值。

1. 以具有管理特权的用户身份登录到控制面板。
2. 选择 **Identity > Projects**。
3. 点击 **Create Project**。
4. 在 **Project Information** 选项卡中，输入项目的名称和描述。**Enabled** 复选框会被默认选中。
5. 在**项目成员**选项卡上，从 **All Users** 列表向项目添加成员。
6. 在 **Quotas** 选项卡上，为项目指定资源限值。
7. 点击 **Create Project**。

6.1.2. 编辑项目

您可以编辑项目以更改其名称或描述、启用或禁用它，或更新项目中的成员。

1. 以具有管理特权的用户身份登录到控制面板。
2. 选择 **Identity > Projects**。
3. 在项目 **Actions** 列中，单击箭头，然后单击 **Edit Project**。
4. 在 **Edit Project** 窗口中，您可以更新项目以更改其名称或描述，并启用或禁用项目。
5. 在 **Project Members** 选项卡上，向项目添加成员，或者根据需要删除它们。
6. 点击 **Save**。

注意

Enabled 复选框会被默认选中。若要临时禁用项目，请清除 **Enabled** 复选框。若要启用禁用的项目，可选中 **Enabled** 复选框。

6.1.3. 删除项目

1. 以具有管理特权的用户身份登录到控制面板。
2. 选择 **Identity > Projects**。
3. 选择您要删除的项目。

4. 单击 **Delete Projects**。此时会显示 **Confirm Delete Projects** 窗口。

5. 单击 **Delete Projects** 以确认操作。

该项目被删除，任何用户对都被解除关联。

6.1.4. 更新项目配额

配额是您为每个项目设置的操作限制，以优化云资源。您可以设置配额以防止项目资源在不通知的情况下耗尽。您可以在项目和 project-user 级别强制配额。

1. 以具有管理特权的用户身份登录到控制面板。

2. 选择 **Identity > Projects**。

3. 在项目 **Actions** 列中，单击箭头，然后单击 **修改配额**。

4. 在 **Quota** 选项卡中，根据需要修改项目配额。

5. 点击 **Save**。

6.1.5. 更改活跃的项目

将项目设置为活跃的项目，以便您可以使用控制面板与项目中的对象交互。要将项目设置为活跃的项目，您必须是项目的成员。用户还需要成为多个项目的成员，才能将 **Set** 设为 **Active Project** 选项。除非重新启用，否则您无法将禁用的项目设置为 active。

1. 以具有管理特权的用户身份登录到控制面板。

2. 选择 **Identity > Projects**。

3. 在项目 **Actions** 列中，单击箭头，然后单击 **Set as Active Project**。

4. 或者，作为非管理员用户，在项目 **Actions** 列中，单击 **Set as Active Project**，它将作为列中的默认操作。

6.2. 项目层次结构

6.2.1. Identity Service 中的分层多租户(HMT)

您可以使用 Identity 服务(keystone)中的多租户来嵌套项目。多租户允许子项目从父项目继承角色分配。

6.2.1.1. 创建项目和子项目

您可以使用 keystone 域和项目实施层次结构多租户(HMT)。首先创建一个新域，然后在该域内创建项目。然后，您可以向该项目添加子项目。您还可以通过将用户添加到该子项目的 **admin** 角色，将用户提升到子项目的管理员。

注意

keystone 使用的 HMT 结构目前没有在仪表板中表示。

1.创建名为 **corp** 的新 keystone 域：

```
$ openstack domain create corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 69436408fdcb44ab9e111691f8e9216d |
| name        | corp                             |
+-------------+----------------------------------+
```

2.在 **corp** 域中创建父项目(**private-cloud**)：

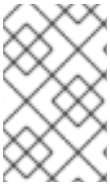
```
$ openstack project create private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | c50d5cf4fe2e4929b98af5abdec3fd64 |
| is_domain   | False                            |
| name        | private-cloud                    |
| parent_id   | 69436408fdcb44ab9e111691f8e9216d |
+-------------+----------------------------------+
```

3.在 **private-cloud** 父项目中创建子项目(**dev**)，同时指定 **corp** 域：

```
$ openstack project create dev --parent private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | 11fccd8369824baa9fc87cf01023fd87 |
| is_domain   | False                            |
| name        | dev                              |
| parent_id   | c50d5cf4fe2e4929b98af5abdec3fd64 |
+-------------+----------------------------------+
```

4.创建名为 **qa** 的另一个子项目：

```
$ openstack project create qa --parent private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | b4f1d6f59ddf413fa040f062a0234871 |
| is_domain   | False                            |
| name        | qa                               |
| parent_id   | c50d5cf4fe2e4929b98af5abdec3fd64 |
+-------------+----------------------------------+
```



注意

您可以使用 Identity API 查看项目层次结构。如需更多信息，请参阅 https://developer.openstack.org/api-ref/identity/v3/index.html?expanded=show-project-details-detail

6.2.1.2. 授予用户访问权限

默认情况下，新创建的项目没有分配的角色。为父项目分配角色权限时，您可以包含 **--inherited** 标志，以指示子项目从父项目中继承分配的权限。例如，具有对父项目的 **admin** 角色的用户也具有对子项目的 **admin** 访问权限。

1.查看分配给项目的现有权限：

```
$ openstack role assignment list --project private-cloud
```

2.查看现有角色：

```
$ openstack role list
+----------------------------------+----------------+
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator  |
| ef3d3f510a474d6c860b4098ad658a29 | service        |
+----------------------------------+----------------+
```

3.授予用户帐户 **user1** 对 **private-cloud** 项目的访问权限：

```
$ openstack role add --user user1 --user-domain corp --project private-cloud member
```

使用 **--inherited** 标志重新运行此命令。因此，**user1** 还可以访问 **private-cloud** 子项目，它已继承了角色分配：

```
$ openstack role add --user user1 --user-domain corp --project private-cloud member --inherited
```

4.查看权限更新的结果：

```
$ openstack role assignment list --effective --user user1 --user-domain corp
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
| Role                             | User                             | Group | Project                          | Domain | Inherited |
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
c50d5cf4fe2e4929b98af5abdec3fd64 |       | False     |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
11fccd8369824baa9fc87cf01023fd87 |       | True      |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
```

```
b4f1d6f59ddf413fa040f062a0234871 |       | True     |
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
```

user1 用户已继承了对 **qa** 和 **dev** 项目的访问权限。此外，由于 **--inherited** 标志应用到父项目，**user1** 也接受对稍后创建的任何子项目的访问。

6.2.2. 删除访问权限

必须单独删除显式和继承的权限。

1.从显式分配的角色中删除用户：

```
$ openstack role remove --user user1 --project private-cloud member
```

2.查看更改的结果。请注意，继承的权限仍然存在：

```
$ openstack role assignment list --effective --user user1 --user-domain corp
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
| Role                             | User                             | Group | Project                          | Domain | Inherited |
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
11fccd8369824baa9fc87cf01023fd87 |       | True     |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
b4f1d6f59ddf413fa040f062a0234871 |       | True     |
+----------------------------------+----------------------------------+-------+----------------------------------+------
--+-----------+
```

3.删除继承的权限：

```
$ openstack role remove --user user1 --project private-cloud member --inherited
```

4.查看更改的结果。继承的权限已被删除，生成的输出现在为空：

```
$ openstack role assignment list --effective --user user1 --user-domain corp
```

6.2.3. 嵌套配额

目前，尚不支持 *嵌套配额*。因此，您必须针对项目和子项目单独管理配额。

6.2.4. 法国概述

使用 *Reseller* 项目时，目标是具有域层次结构；这些域最终允许您考虑删除部分云，子域代表一个功能齐全的云。这个工作分为几个阶段，如下所示：

6.2.4.1. 1阶段

法国（第1阶段）是层次结构多租户(HMT)的扩展，如下所述： 第 6.2.1 节 “Identity Service 中的分层多租户(HMT)”。在以前的版本中，keystone 域最初打算是存储用户和项目的容器，在数据库后端中都有自己的表。现在，域不再存储在自己的表中，并已合并到项目表中：

- 域现在是一个项目，由 **is_domain** 标志区分。
- 域表示项目层次结构中的顶级项目：域是项目层次结构中的根
- API 已更新，以使用 **项目** 子路径来创建和检索域：
  - 通过创建项目并将 **is_domain** 标志设置为 true 来创建新域
  - 列出属于 domain 的项目：获取项目，包括 **is_domain** 查询参数。

~

6.3. 项目安全管理

安全组是可分配给项目实例的 IP 过滤器规则集合，用于定义对实例的网络访问。安全组特定于项目；项目成员可以编辑其安全组的默认规则，并添加新规则集。

所有项目都有一个默认安全组，应用到没有其他定义的安全组的任何实例。除非更改默认值，否则此安全组会拒绝所有传入的流量，并且只允许来自您的实例的传出流量。

您可以在实例创建过程中直接将安全组应用到实例，或应用到正在运行的实例上的端口。

注意

您不能在实例创建过程中将基于角色的访问控制(RBAC)共享安全组直接应用到实例。要将 RBAC 共享安全组应用到实例，必须首先创建端口，将共享安全组应用到该端口，然后将该端口分配给实例。请参阅 将安全组添加到端口。

在不创建允许所需出口的组的情况下，不要删除默认安全组。例如，如果您的实例使用 DHCP 和元数据，您的实例需要安全组规则来允许到 DHCP 服务器和元数据代理的出口。

6.3.1. 创建安全组

1. 在仪表板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡中，单击 **Create Security Group**。
3. 输入组的名称和描述，然后单击 **Create Security Group**。

6.3.2. 添加安全组规则

默认情况下，新组的规则仅提供传出访问。您必须添加新规则来提供其他访问权限。

1. 在仪表板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡中，点您要编辑的安全组的**管理规则**。
3. 单击 **Add Rule** 以添加新规则。
4. 指定规则值，然后单击 **Add**。
   以下规则字段是必需的：

   规则

   规则类型。如果您指定一个规则模板（例如，*SSH*），其字段会自动填写：

- TCP：用来在系统之间交换数据，以及用于最终用户通信。
- UDP：用来在系统间交换数据，特别是在应用程序级别。
- ICMP：由网络设备（如路由器）使用来发送错误或监控消息。

方向

Ingress （入站）或 Egress （出站）。

打开端口

对于 TCP 或 UDP 规则，打开 **Port** 或 **Port Range**（单个端口或端口范围）：

- 对于端口范围，在 **From Port** 和 **To Port** 字段中输入 port 值。
- 对于单个端口，在 **Port** 字段中输入 port 值。

类型

ICMP 规则的类型；必须在 *-1:255* 之间。

代码

ICMP 规则的代码；必须在 *-1:255* 之间。

远程

此规则的流量源：

- CIDR (Classless Inter-Domain Routing)：IP 地址块，限制对块内 IP 的访问。在 Source 字段中输入 CIDR。
- 安全组：源组，使组中的任何实例能够访问任何其他组实例。

6.3.3. 删除安全组规则

1. 在仪表板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡上，点安全组的管理规则。
3. **选择安全组规则，然后点删除规则。**
4. **再次点 Delete Rule。**

注意

您无法撤销 delete 操作。

6.3.4. 删除安全组

1. **在仪表板中，选择 Project > Compute > Access & Security。**
2. **在 Security Groups 选项卡中，选择组，然后单击 Delete Security Groups。**
3. **单击 Delete Security Groups。**

注意

您无法撤销 delete 操作。

第 7 章 管理域

Identity Service (keystone)域是您可以在 keystone 中创建的额外命名空间。使用 keystone 域对用户、组和项目进行分区。您还可以配置这些独立域，以便在不同的 LDAP 或 Active Directory 环境中验证用户身份。如需更多信息，请参阅 与 Identity Service 集成指南。

注意

Identity Service 包括一个名为 **Default** 的内置域。建议您只为服务帐户保留这个域，并为用户帐户创建单独的域。

7.1. 查看域列表

您可以使用 openstack domain list 查看域列表：

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

7.2. 创建新域

您可以使用 openstack domain create 创建新域：

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

7.3. 查看域的详情

您可以使用 openstack domain show 查看域的详情：

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

7.4. 禁用域

1. **您可以使用 --disable 禁用域：**

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

2. **确认域已被禁用：**

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

3. **然后，如果需要可以重新启用域：**

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

第 8 章 身份管理

8.1. 安全 LDAP 通信

如果您已将 Identity 服务(keystone)配置为针对或从 LDAP 服务器检索身份信息，您可以使用 CA 证书保护身份服务的 LDAP 通信。

您必须从 Active Directory 获取 CA 证书，将 CA 证书文件转换为 Privacy Enhanced Mail (PEM)文件格式，并为 Identity 服务配置安全 LDAP 通信。您可以根据配置 CA 信任的位置和方式，通过三种方法之一执行此配置。

8.1.1. 从 Active Directory 获取 CA 证书

使用以下示例代码查询 Active Directory 以获取 CA 证书。CA_NAME 是证书的名称，并可根据您的配置更改其余参数：

```
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
```

8.1.2. 将 CA 证书转换为 PEM 文件格式

创建名为 /path/cacert.pem 的文件，并包含从 Active Directory 获取 CA 证书的 LDAP 查询站的内容，该文件在标头和页脚内：

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

为进行故障排除，您可以执行以下查询来检查 LDAP 是否正常工作，并确保 PEM 证书文件是否已正确创建。

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

查询应返回类似如下的结果：

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

如果 CA 证书由 web 服务器托管，您可以运行以下命令获取 CA 证书。

Example

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

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

8.1.3. 为 Identity 服务配置安全 LDAP 通信的方法

8.1.3.1. 方法 1

如果使用 PEM 文件在 LDAP 级别配置了 CA 信任，请使用此方法。手动指定 CA 证书文件的位置。以下流程保护 LDAP 通信不仅针对 Identity 服务的安全，而不适用于所有使用 OpenLDAP 库的应用程序。

1. **将包含 PEM 格式的 CA 证书链的文件复制到 /etc/openldap/certs 目录。**
2. **编辑 /etc/openldap/ldap.conf 并添加以下指令，将 [CA_FILE] 替换为 CA 证书文件的位置和名称：**

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

3. **重启 horizon 容器：**

   ```
   # systemctl restart tripleo_horizon
   ```

8.1.3.2. 方法 2

如果使用网络安全服务(NSS)数据库在 LDAP 库级别配置了 CA 信任，则使用此方法。使用 certutil 命令将 CA 证书导入并信任到 OpenLDAP 库使用的 NSS 证书数据库中。以下流程保护 LDAP 通信不仅针对 Identity 服务的安全，而不适用于所有使用 OpenLDAP 库的应用程序。

1. **导入并信任证书，将 [CA_FILE] 替换为 CA 证书文件的位置和名称：**

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

2. **确认 CA 证书是否已正确导入：**

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

   您的 CA 证书已被列出，信任属性被设置为 CT,。

3. **重启 horizon 容器：**

   ```
   # systemctl restart tripleo_horizon
   ```

8.1.3.3. 方法 3

如果使用 PEM 文件在 Keystone 级别配置了 CA 信任，则使用此方法。保护 Identity 服务和 LDAP 服务器之间的通信的最终方法是为 Identity 服务配置 TLS。

但是，与以上两种方法不同，此方法只为 Identity 服务保护 LDAP 通信，且不会保护使用 OpenLDAP 库的其他应用程序的 LDAP 通信。

以下流程使用 **openstack-config** 命令编辑 **/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf** 文件中的值。

1. 启用 TLS：

```
# openstack-config --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf ldap use_tls True
```

2. 指定证书的位置，将 [CA_FILE] 替换为 CA 证书的名称：

```
# openstack-config --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf ldap tls_cacertfile [CA_FILE]
```

3. 指定 LDAP 服务器的传入 TLS 会话上执行的客户端证书检查，将 [CERT_BEHAVIOR] 替换为以下列出的行为之一：

   需求

   从 LDAP 服务器请求证书。如果没有提供证书，或者无法针对现有证书颁发机构文件验证所提供的证书，则会话将被终止。

   allow

   从 LDAP 服务器请求证书。即使未提供证书，会话也会正常进行。如果提供了证书，但无法针对现有的证书颁发机构文件进行验证，则证书将被忽略，会话将正常运行。

   never

   将不会请求证书。

```
# openstack-config --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf ldap tls_req_cert [CERT_BEHAVIOR]
```

4. 重启 keystone 和 horizon 容器：

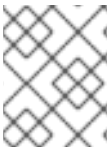
```
# systemctl restart tripleo_keystone
# systemctl restart tripleo_horizon
```

第 9 章 应用程序凭证

使用 *应用凭据* 以避免在配置文件中嵌入用户帐户凭据。相反，用户会创建一个应用程序凭据，它接受对单个项目委托的访问权限，并具有自己的不同机密。用户也可以将委派的特权限制为该项目中的单个角色。这可让您采用最小特权的原则，其中经过身份验证的用户只能访问一个项目和角色，而不是所有项目和角色。

您可以使用此方法在不显示用户凭据的情况下使用 API，应用可以向 Keystone 进行身份验证，而无需嵌入式用户凭据。

您可以使用应用程序凭证来生成令牌，并为应用程序配置 keystone_authtoken 设置。在以下部分中描述了这些用例。



注意

应用程序凭据取决于创建它的用户帐户，因此如果该帐户被删除，或者丢失对相关角色的访问，它将终止。

9.1. 使用应用程序凭证生成令牌

应用凭据可在仪表板中作为自助服务功能供用户使用。本例演示了如何创建应用程序凭证，然后使用它生成令牌。

1. 创建测试项目并测试用户帐户：

   a. 创建名为 **AppCreds** 的项目：

   ```
   $ openstack project create AppCreds
   ```

   b. 创建名为 **AppCredsUser** 的用户：

   ```
   $ openstack user create --project AppCreds --password-prompt AppCredsUser
   ```

   c. 为 **AppCreds** 项目授予 **AppCredsUser** 角色的访问权限：

   ```
   $ openstack role add --user AppCredsUser --project AppCreds member
   ```

2. 以 **AppCredsUser** 用户身份登录仪表板并创建应用程序凭证：

   概述 → Identity → Application Credentials → +Create Application Credential。

   注意

   确保您下载 **clouds.yaml** 文件内容，因为在关闭应用程序凭证的弹出窗口后，您无法再次访问它。

3.

使用 CLI 创建名为 /home/stack/.config/openstack/clouds.yaml 的文件并粘贴 clouds.yaml 文件的内容。

```
# This is a clouds.yaml file, which can be used by OpenStack tools as a source
# of configuration on how to connect to a cloud. If this is your only cloud,
# just put this file in ~/.config/openstack/clouds.yaml and tools like
# python-openstackclient will just work with no further config. (You will need
# to add your password to the auth section)
# If you have more than one cloud account, add the cloud entry to the clouds
# section of your existing file and you can refer to them by name with
# OS_CLOUD=openstack or --os-cloud=openstack
clouds:
  openstack:
    auth:
      auth_url: http://10.0.0.10:5000/v3
      application_credential_id: "6d141f23732b498e99db8186136c611b"
      application_credential_secret: "<example secret value>"
    region_name: "regionOne"
    interface: "public"
    identity_api_version: 3
    auth_type: "v3applicationcredential"
```

注意

对于部署，这些值会有所不同。

4. 使用应用凭据生成令牌。在使用以下命令时，不得以任何特定用户形式提供，且您必须与 clouds.yaml 文件位于同一个目录中。

```
[stack@undercloud-0 openstack]$ openstack --os-cloud=openstack token issue
+------------+------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------+
| Field      | Value
|
+------------+------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------+
| expires    | 2018-08-29T05:37:29+0000
|
| id         | gAAAAABbhiMJ4TxxFlTMdsYJpfStsGotPrns0lnpvJq9ILtdi-
NKqisWBeNiJlUXwmnoGQDh2CMyK9OeTsuEXnJNmFfKjxiHWmcQVYzAhMKo6_QMUtu_Qm
6mtpzYYHBrUGboa_Ay0LBuFDtsjtgtvJ-r8G3TsJMowbKF-yo--
O_XLhERU_QQVl3hl8zmMRdmLh_P9Cbhuolt |
| project_id | 1a74eabbf05c41baadd716179bb9e1da
|
| user_id    | ef679eeddfd14f8b86becfd7e1dc84f2
|
+------------+------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------+
```

注意

如果您收到类似于 __init__（）的错误，则会出现一个意外的关键字参数 'application_credential_secret'，那么您可能仍会 source 到之前的凭证。对于全新环境，请运行 sudo su - stack。

9.2. 将应用程序凭证与应用程序集成

应用程序凭证可用于向 keystone 验证应用程序。当使用应用程序凭证时，keystone_authtoken 设置使用 v3applicationcredential 作为身份验证类型，并包含您在凭证创建过程中收到的凭证。输入以下值：

- application_credential_secret ：应用程序凭证 secret。
- application_credential_id ：应用程序凭证 ID。
- （可选） application_credential_name ：如果您使用命名的应用程序凭证，而不是 ID，则可以使用此参数。

例如：

```
[keystone_authtoken]
auth_url = http://10.0.0.10:5000/v3
auth_type = v3applicationcredential
application_credential_id = "6cb5fa6a13184e6fab65ba2108adf50c"
application_credential_secret = "<example password>"
```

9.3. 使用命令行管理应用程序凭证

您可以使用命令行来创建和删除应用凭证。

create 子命令基于当前源的帐户创建一个应用程序凭据。例如，当作为 admin 用户提供凭证时创建凭证，会将同一角色授予应用程序凭证：

```
$ openstack application credential create --description "App Creds - All roles" AppCredsUser
+--------------+-----------------------------------------------------------------------------------+
| Field        | Value                                                                             |
+--------------+-----------------------------------------------------------------------------------+
```

```
| description  | App Creds - All roles                                                                                |
| expires_at   | None                                                                                                 |
| id           | fc17651c2c114fd6813f86fdbb430053                                                                     |
| name         | AppCredsUser                                                                                         |
| project_id   | 507663d0cfe244f8bc0694e6ed54d886                                                                     |
| roles        | member reader admin                                                                                  |
| secret       | fVnqa6I_XeRDDkmQnB5lx361W1jHtOtw3ci_mf_tOID-09MrPAzkU7mv-
by8ykEhEa1QLPFJLNV4cS2Roo9lOg |
| unrestricted | False                                                                                                |
+--------------+------------------------------------------------------------------------------------------------------+
```

警告

使用 --unrestricted 参数可让应用程序凭证创建和删除其他应用程序凭证和信任。这可能是危险的行为，默认是禁用的。您不能将 --unrestricted 参数与其他访问规则结合使用。

默认情况下，生成的角色成员资格包括分配给创建凭据的帐户的所有角色。您可以把访问权限委派给特定角色来限制角色成员资格：

```
$ openstack application credential create --description "App Creds - Member" --role member
AppCredsUser
+--------------+------------------------------------------------------------------------------------------------------+
| Field        | Value                                                                                                |
+--------------+------------------------------------------------------------------------------------------------------+
| description  | App Creds - Member                                                                                   |
| expires_at   | None                                                                                                 |
| id           | e21e7f4b578240f79814085a169c9a44                                                                     |
| name         | AppCredsUser                                                                                         |
| project_id   | 507663d0cfe244f8bc0694e6ed54d886                                                                     |
| roles        | member                                                                                               |
| secret       |
XCLVUTYIreFhpMqLVB5XXovs_z9JdoZWpdwrkaG1qi5GQcmBMUFG7cN2htzMlFe5T5mdPsnf5JMNb
u0Ih-4aCg |
| unrestricted | False                                                                                                |
+--------------+------------------------------------------------------------------------------------------------------+
```

删除应用程序凭证：

```
$ openstack application credential delete AppCredsUser
```

9.4. 操作任务

9.4.1. 替换现有的应用程序凭证

应用程序凭据绑定到创建它们的用户帐户，如果用户帐户被删除，或者用户丢失对委派的角色的访问权限，则无效。因此，您应该准备好根据需要生成新的应用程序凭证。

9.4.2. 对于配置文件

更新分配给应用程序的应用程序凭据（使用配置文件）：

1. 创建新应用程序凭证集合。
2. 将新凭据添加到应用程序配置文件中，替换现有的凭据。更多信息请参阅 第 9.2 节 “将应用程序凭证与应用程序集成”。
3. 重新启动应用服务以应用更改。
4. 删除旧应用凭据（如果适用）。有关命令行选项的详情请参考 第 9.3 节 “使用命令行管理应用程序凭证”。

9.4.3. 对于 clouds.yaml 文件

替换 clouds.yaml 使用的现有应用程序凭证：

例如，如果您的 clouds.yaml 包含一个名为 AppCred1 的应用程序凭证，则过期：

1. 创建名为 *AppCred2* 的应用凭据。
2. 在删除 AppCred1 配置时，将新的 AppCred2 添加到 clouds.yaml 文件中。
3. 使用 clouds.yaml 生成令牌，以确认凭证是否按预期工作。如需更多信息，请参阅 第 9.1 节 “使用应用程序凭证生成令牌” 的第 4 步。