Red Hat OpenStack Platform 17.1

管理 OpenStack Identity 资源

配置用户和 keystone 身份验证

Last Updated: 2024-05-23

Red Hat OpenStack Platform 17.1 管理 OpenStack Identity 资源

配置用户和 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

您无法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，您必须首先创建端口，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 向端口 添加安全组。

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 的信息。

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。

在 JIRA 中提供文档反馈

使用 Create Issue 表单对文档提供反馈。JIRA 问题将在 Red Hat OpenStack Platform Jira 项目中创建，您可以在其中跟踪您的反馈进度。

1. 确保您已登录到 JIRA。如果您没有 JIRA 帐户，请创建一个帐户来提交反馈。
2. 点击以下链接打开 **Create Issue** 页面： Create Issue
3. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节或章节号以及问题的详细描述。不要修改表单中的任何其他字段。
4. 点 **Create**。

第 1 章 IDENTITY SERVICE (KEYSTONE)简介.

作为云管理员，您可以管理项目、用户和角色。

项目是包含资源集合的组织单元。您可以将用户分配到项目中的角色。角色定义用户可以对给定项目中的资源执行的操作。可以在多个项目中为用户分配角色。

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

- 添加、更新和删除项目与用户。
- 将用户分配到一个或多个角色，并更改或删除这些分配。
- 相互独立管理项目和用户。

您还可以使用身份服务(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，以便您可以从外部管理用户和身份，并将用户数据与身份服务同步。

1.1. 资源凭据文件

安装 Red Hat OpenStack Platform director 时，会自动生成资源凭证(RC)文件：

```
# Clear any old environment that may conflict.
for key in $( set | awk -F= '/^OS_/ {print $1}' ); do unset "${key}" ; done
export OS_CLOUD=undercloud
# Add OS_CLOUDNAME to PS1
if [ -z "${CLOUDPROMPT_ENABLED:-}" ]; then
    export PS1=${PS1:-""}
    export PS1=\${OS_CLOUD:+"(\$OS_CLOUD)"}\ $PS1
    export CLOUDPROMPT_ENABLED=1
fi
export PYTHONWARNINGS="ignore:Certificate has no, ignore:A true SSLContext object is not available"
```

提供 **stackrc** 文件，将身份验证详情导出到 shell 环境中。这可让您针对本地 Red Hat OpenStack Platform director API 运行命令。

在安装 overcloud 期间生成的 RC 文件的名称是部署的堆栈的名称，使用 'rc' 后缀。如果没有为您的堆栈提供自定义名称，则堆栈被标记为 **overcloud**。创建称为 **overcloudrc** 的 RC 文件：

```
# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="}  /^OS_/ {print $1}' ); do unset $key ; done
export OS_USERNAME=admin
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_NO_CACHE=True
export OS_CLOUDNAME=overcloud
export no_proxy=10.0.0.145,192.168.24.27
export PYTHONWARNINGS='ignore:Certificate has no, ignore:A true SSLContext object is not available'
export OS_AUTH_TYPE=password
export OS_PASSWORD=mpWt4y0Qhc9oTdACisp4wgo7F
```

```
export OS_AUTH_URL=http://10.0.0.145:5000
export OS_IDENTITY_API_VERSION=3
export OS_COMPUTE_API_VERSION=2.latest
export OS_IMAGE_API_VERSION=2
export OS_VOLUME_API_VERSION=3
export OS_REGION_NAME=regionOne

# Add OS_CLOUDNAME to PS1
if [ -z "${CLOUDPROMPT_ENABLED:-}" ]; then
    export PS1=${PS1:-""}
    export PS1=\${OS_CLOUDNAME:+"(\$OS_CLOUDNAME)"}\ $PS1
    export CLOUDPROMPT_ENABLED=1
fi
```

overcloud RC 文件在文档中被称为 **overcloudrc**，无论堆栈的实际名称是什么。提供 **overcloudrc** 文件，将身份验证详细信息导出到 shell 环境中。这可让您针对 overcloud 集群的 control plane API 运行命令。自动生成的 **overcloudrc** 文件会在 **admin** 项目中将您验证为 **admin** 用户。此身份验证对域管理任务（如创建提供商网络或项目）有价值。

1.2. OPENSTACK 区域

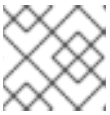
区域是 OpenStack 部署的一个划分。每个区域都有自己的完整的 OpenStack 部署，包括自己的 API 端点、网络和计算资源。不同的地区共享一组身份服务(keystone)和 Dashboard 服务(horizon)服务，以提供访问控制和 Web 界面。Red Hat OpenStack Platform 使用单一区域进行部署。默认情况下，您的 overcloud 区域命名为 **regionOne**。您可以更改 Red Hat OpenStack Platform 中的默认区域名称。

流程

- 在 **parameter_defaults** 下，定义 **KeystoneRegion** 参数：

```
parameter_defaults:
  KeystoneRegion: '<sample_region>'
```

- 将 **<sample_region** > 替换为您选择的区域名称。



注意

在部署 overcloud 后，您无法修改区域名称。

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。您可以相互独立管理项目和用户。

2.1. 使用仪表板创建用户

您可以为用户分配主要项目和角色。使用 OpenStack Dashboard (horizon)创建的用户默认为 Identity 服务用户。您可以通过配置包含在 Identity 服务的 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. 点 **Create User**.
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 **成**员。
7. 点 **Create User**.

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，单击 **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。与删除用户不同，此操作不可逆。

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。

结果是您禁用了的用户无法：

- 登录控制面板。

- 访问 RHOSP 服务。
- 在控制面板中执行任何 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，值随后更新为 **True** 或 **False**。

2.4. 使用仪表板删除用户

您必须是一个具有管理角色的用户，才能删除其他用户。此操作无法撤销。

流程

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。角色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。
- 启用非管理员用户管理环境的管理角色。

注意

Identity 服务(keystone)也添加了 **reader** 角色，该角色将显示在角色列表中。只有在启用了安全 RBAC 时，才使用 **reader** 角色。

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 ADMIN 角色

当您为用户分配 **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** 角色的用户可以创建或删除用户并管理角色。

为您为用户分配 **admin** 角色的项目是执行 **openstack** 命令的默认项目。例如，如果一个 **admin** 用户在一个名为 **development** 的项目中运行以下命令，将会在名为 **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
```

示例

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

注意

要获得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。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，您可以使用身份服务(keystone)客户端通过以下一组命令来创建和管理角色：每个 Red Hat OpenStack Platform 部署都必须包括至少一个项目、一个用户和一个角色链接在一起。

您可以将用户分配到多个项目。要将用户分配到多个项目，请创建一个角色，并将该角色分配给用户项目对。

注意

您可以使用名称或 ID 来指定用户、角色或项目。

流程

1. 创建一个 **new-role** 角色：

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

2. 要为项目分配用户，请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. 将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** 角色是否已将 **admin** 用户分配给 **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. 创建简化的角色

Identity 服务(keystone)强制执行访问控制，确认用户已分配给特定的角色。Identity 服务使用 implied 角色分配。如果您明确为用户分配角色，那么用户也可以隐式分配给其他角色。您可以在 Red Hat OpenStack Platform 中查看默认含义的角色：

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

注意

Identity 服务(keystone)也添加了 **reader** 角色，该角色将显示在角色列表中。只有在启用了安全 RBAC 时，才使用 **reader** 角色。

具有较高权限的角色意味着与角色关联的权限较少。在上面的默认代表角色中，admin 表示成员，成员代表读者。通过简化角色，用户的角色分配会累计处理，以便用户继承下级角色。

如果使用自定义角色，您可以创建简化的关联。

注意

当您创建新角色时，它默认具有与 **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

- 列出所有简化的角色：

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

如果简化关联出错，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 CLI 配置组

创建组并为组分配权限。组成员继承分配给组的相同权限：

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. 授予 **grp-Auditors** 组权限来访问 **demo** 项目，同时使用 **member** 角色：

```
$ 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. 查看已分配给 **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. 使用仪表板配置组

您可以使用控制面板管理 keystone 组成员身份。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 CLI 配置组。

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管理 keystone 组成员身份。

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** 向组添加用户。如果要删除用户，选中其复选框并点 **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。每个项目都会被分配资源，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，而不会相互干扰权限和资源。在创建新项目时，会预先配置一组资源配额。配额包括可分配给项目的 VCPU 数、实例、RAM 和浮动 IP 数。配额可以在项目和项目用户级别上强制执行。您可以使用控制面板为新的和现有项目设置或修改 Compute 和 Block Storage 配额。如需更多信息，请参阅管理项目。

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。

流程

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

示例

```
$ nova quota-show --user 3b9763e4753843529db15085874b1e84 --tenant
a4ee0cbb97e749dca6de584c0b1568a6
+-----------------------------+-------+
| 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-ID] --[QUOTA_NAME] [QUOTA_VALUE] [TENANT-ID]
$ nova quota-show --user [USER-ID] --tenant [TENANT-ID]
```

示例

```
$ nova quota-update --user 3b9763e4753843529db15085874b1e84 --floating-ips 10
a4ee0cbb97e749dca6de584c0b1568a6
$ nova quota-show --user 3b9763e4753843529db15085874b1e84 --tenant
a4ee0cbb97e749dca6de584c0b1568a6
```

```
+----------------------------+-------+
| 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，Object Storage 代理服务器必须具有参数 **container_quotas** 或 **account_quotas** （或两者）添加到 **proxy-server.conf** 文件的 **[pipeline:main]** 部分：

```
[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 配额放在一个帐户中：

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

第 6 章 管理项目

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，您可以为其分配 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，这些项目不会相互干扰权限和资源。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. 创建一个项目

创建一个项目，向项目添加成员，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**。

6.2. 编辑项目

您可以编辑项目来更改其名称或描述、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控制面板。
2. 选择 **Identity > Projects**。
3. 在项目 **Actions** 列中，单击箭头，再单击 **Edit Project**。
4. 在 **Edit Project** 窗口中，您可以更新一个项目以更改其名称或描述，并启用或禁用项目。
5. 在 **项目成员选项卡** 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，可清除 **Enabled** 复选框。要启用禁用的项目，请选中 **Enabled** 复选框。

6.3. 删除项目

1. 以具有管理特权的用户身份登录控制面板。
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。

该项目已被删除，任何用户对均将被解除关联。

6.4. 更新项目配额

配额是您要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没有通知的情况下耗尽。您可以在项目和项目用户级别强制实施配额。

1. 以具有管理特权的用户身份登录控制面板。
2. 选择 **Identity > Projects**。
3. 在项目 **Actions** 列中，单击箭头，再单击 **Modify Quotas**。
4. 在 **Quota** 选项卡中，根据需要修改项目配额。
5. 点击 **Save**。

注意

目前还不支持 *嵌套配额*。因此，您必须针对项目和子项目单独管理配额。

6.5. 更改活跃的项目

将项目设置为活动项目，以便您可以使用控制面板与项目中的对象交互。要将项目设置为活动项目，您必须是项目的成员。用户还需要成为多个项目的成员，以便启用 **Set 作为 Active Project** 选项。除非被重新启用，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控制面板。
2. 选择 **Identity > Projects**。
3. 在项目 **Actions** 列中，单击箭头，再单击 **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**，它将成为列中的默认操作。

6.6. 项目层次结构

您可以使用身份服务(keystone)中的多租户项目嵌套项目。多租户允许子项目从父项目继承角色分配。

6.6.1. 创建分层项目和子项目

您可以使用 keystone 域和项目实施层次结构多租户(HMT)。首先创建新域，然后在该域中创建项目。然后，您可以将子项目添加到该项目。您还可以通过将用户添加到该子项目的 **admin** 角色，将用户提升为子项目的管理员。

注意

keystone 使用的 HMT 结构目前没有在仪表板中表示。

流程

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

6.6.2. 配置对分层项目的访问

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。

授予用户访问权限

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** 也接收对稍后创建的任何子项目的访问权限。

从用户中删除访问

必须单独删除显式和继承的权限。

1. 从显式分配角色中删除用户：

```
$ 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，结果输出现在为空：

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

6.6.3. 经销商项目概述

使用 *Reseller* 项目时，目标是拥有域层次结构；这些域最终允许您考虑转销部分云，其带有代表完全启用云的子域。这个工作分为几个阶段，阶段 1 如下所示：

经销商阶段 1

经销商（阶段 1）是层次结构多租户(HMT)的扩展，如下所述： 创建分层项目和子项目。在以前的版本中，keystone 域最初设计为是存储用户和项目的容器，其具有他们在数据库后端中的表。因此，域现在不再存储在自己的表中，并已合并到项目表中：

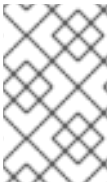
- 域现在是项目的类型，可通过 **is_domain** 标志区分。
- 域代表项目层次结构中的顶级项目：域是项目层次结构中的 roots
- API 已更新，以使用 **项目** 子路径创建和检索域：
  - 通过创建将 **is_domain** 标志设为 true 的项目来创建新域
  - 列出 domain: get projects，包括 **is_domain** 查询参数。

6.7. 项目安全管理

安全组是 IP 过滤规则的集合，可以分配给项目实例，以及定义实例的网络访问。安全组是特定于项目的；项目成员可以编辑其安全组的默认规则，并且添加新的规则集。

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。除非更改了默认值，否则此安全组拒绝所有传入流量，并且仅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

您无法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，您必须首先创建端口，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 向端口 添加安全组。

不要在不创建允许所需出口的组的情况下删除默认安全组。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。

6.7.1. 创建安全组

创建安全组，以便您可以配置安全规则。例如，您可以启用 ICMP 流量，或者禁用 HTTP 请求。

流程

1. 在控制面板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡中，点 **Create Security Group**。
3. 输入组的名称和描述，再单击 **Create Security Group**。

6.7.2. 添加安全组规则

默认情况下，新组的规则仅提供传出访问。您必须添加新规则才能提供额外的访问。

流程

1. 在控制面板中，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，然后点 **Add**。
以下规则字段是必需的：

规则

规则类型。如果您指定了规则模板（如 'SSH'），则会自动填写其字段：

- TCP：通常用来在系统之间交换数据，以及用于最终用户通信。
- UDP：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 (inbound)或 Egress (outbound)。

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）：

- 对于一系列端口，在 **From Port** 和 **To Port** 字段中输入端口值。
- 对于单个端口，在 **Port** 字段中输入端口值。

类型

ICMP 规则的类型；必须在范围 '-1:255' 中。

代码

ICMP 规则的代码；必须在范围 '-1:255' 中。

远程

此规则的流量源：

- CIDR (Classless Inter-Domain Routing)：IP 地址块，限制对块中 IP 的访问。在 Source 字段中输入 CIDR。
- 安全组：允许组中的任何实例访问任何其他组实例的 Source 组。

6.7.3. 删除安全组规则

删除您不再需要的安全组规则。

流程

1. 在控制面板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡中，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. 再次单击 **Delete Rule**。

注意

您不能撤销删除操作。

6.7.4. 删除安全组

删除您不再需要的安全组。

流程

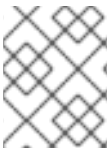
1. 在控制面板中，选择 **Project > Compute > Access & Security**。
2. 在 **Security Groups** 选项卡中，选择组，然后单击 **Delete Security Groups**。
3. 单击 **Delete Security Groups**。

注意

您不能撤销删除操作。

第 7 章 管理域

Identity Service (keystone)域是您可以在 keystone 中创建的其他命名空间。使用 keystone 域对用户、组和项目进行分区。您还可以配置这些单独的域来验证不同 LDAP 或 Active Directory 环境中的用户。如需更多信息，请参阅 集成 Identity Service 指南。



注意

Identity Service 包含一个名为 **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** 命令查看域列表：

```
$ 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** 命令创建新域：

```
$ 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** 命令查看域的详情：

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

7.4. 禁用域

您可以根据要求禁用并启用域。

流程

1. 使用 **--disable** 选项禁用域：

```
$ 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. 如果需要，使用 **--enable** 选项重新启用域：

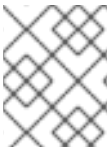
```
$ openstack domain set TestDomain --enable
```

第 8 章 应用程序凭证

使用 *应用凭据* 来避免将用户帐户凭据嵌入到配置文件中。相反，用户会创建一个 Application Credential，它接收委派对单个项目的访问权限，并具有自己的不同的机密。用户也可以将委派的特权限制为该项目中的单个角色。这样，您可以采用最小特权的原则，其中经过身份验证的用户只能获得一个项目以及需要正常工作的角色，而不是所有项目和角色。

您可以使用这种方法来消耗 API，而不显示用户凭据，应用可以在不需要嵌入的用户凭据的情况下向 Keystone 进行身份验证。

您可以使用应用程序凭证为应用程序生成令牌并配置 **keystone_authtoken** 设置。以下部分描述了这些用例。



注意

Application Credential 依赖于创建它的用户帐户，因此当该帐户被删除或丢失对相关角色的访问权限时，它将终止。

8.1. 使用应用程序凭证生成令牌

应用程序凭据在仪表板中作为自助服务功能提供给用户使用。本例演示了用户如何创建应用凭据，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
   ```

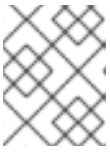
   b. 创建名为 **AppCredsUser** 的用户：

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

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

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

2. 以 **AppCredsUser** 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，因为在关闭了 **Your Application Credential** 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

您的部署这些值将有所不同。

4. 使用 Application Credential 来生成令牌。使用以下命令时，不得作为任何特定用户提供，且必须与 **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

如果您收到与 ***init***（）类似的错误，则获取 **unexpected** 关键字参数 **'application_credential_secret'**，则您可能仍会 source 到前面的凭证。对于全新的环境，请运行 **sudo su - 堆栈**。

8.2. 将应用程序凭证与应用程序集成

应用凭据可用于向 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接收的凭证。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，您可以使用此参数。

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>"
```

8.3. 管理应用凭证

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凭据。例如，当以 **admin** 用户身份提供时创建凭证会将相同的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。这是潜在的危险行为，默认是禁用的。您不能与其他访问规则结合使用 **--unrestricted** 参数。

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。您可以通过将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
```

8.4. 替换应用程序凭证

应用凭据将绑定到创建它们的用户帐户，并在用户帐户被删除时无效，或者用户丢失对委派的角色的访问权限。因此，您应该准备根据需要生成新应用程序凭证。

替换配置文件的现有应用程序凭证

更新分配给应用程序的应用程序凭证（使用配置文件）：

1. 创建一组新的应用凭据。
2. 将新凭据添加到应用配置文件中，替换现有的凭据。如需更多信息，请参阅将应用程序凭证与应用程序集成。
3. 重启应用程序服务以应用更改。
4. 删除旧应用程序凭证（如果适用）。有关命令行选项的更多信息，请参阅管理应用程序凭证。

替换 clouds.yaml 中的现有应用程序凭证

替换 **clouds.yaml** 使用的应用程序凭证时，您必须使用 OpenStack 用户凭证创建替换凭证。默认情况下，您无法使用应用程序凭证来创建另一组应用程序凭证。**openstack application credential create** 命令基于当前源的帐户创建一个应用程序凭证。

1. 以 OpenStack 用户身份进行身份验证，后者最初创建即将过期的身份验证凭据。例如，如果您使用应用程序凭证 生成令牌的步骤， 则必须再次以 **AppCredsUser** 身份登录。
2. 创建名为 **AppCred2** 的应用凭据。这可以通过 OpenStack Dashboard 或 **openstack** CLI 界面完成：

   ```
   openstack application credential create --description "App Creds 2 - Member" --role member
   AppCred2
   ```

3. 复制上一命令输出中的 **id** 和 **secret** 参数。**secret** 参数值无法再次访问。
4. 将 **${HOME}/.config/openstack/clouds.yaml** 文件中的 **application_credential_id** 和 **application_credential_secret** 参数值替换为您复制的 **secret** 和 **id** 值。

验证

1. 使用 clouds.yaml 生成令牌，以确认凭证按预期工作。使用以下命令时，不得作为任何特定用户提供，且必须与 **clouds.yaml** 文件位于同一个目录中：

```
[stack@undercloud-0 openstack]$ openstack --os-cloud=openstack token issue
```

输出示例：

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