Red Hat OpenStack Platform 17.1

OpenStack Identity リソースの管理

ユーザーと keystone 認証の設定

Last Updated: 2024-08-12

Red Hat OpenStack Platform 17.1 OpenStack Identity リソースの管理

ユーザーと keystone 認証の設定

OpenStack Team
rhos-docs@redhat.com

法律上の通知

概要

アプリケーション認証情報、ユーザー、ロール、プロジェクト、およびクォータを管理します。

目次

はじめに

注記

インスタンスの作成中に、ロールベースのアクセス制御 (RBAC) 共有セキュリティーグループをインスタンスに直接適用することはできません。RBAC 共有セキュリティーグループをインスタンスに適用するには、最初にポートを作成し、共有セキュリティーグループをそのポートに適用してから、そのポートをインスタンスに割り当てる必要があります。セキュリティーグループのポートへの追加 を参照してください。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Jira でドキュメントのフィードバックを提供する

ドキュメントに関するフィードバックを提供するには、Create Issue フォームを使用します。Red Hat OpenStack Platform Jira プロジェクトで Jira Issue が作成され、フィードバックの進行状況を追跡できます。

1. Jira にログインしていることを確認してください。Jira アカウントをお持ちでない場合は、アカウントを作成してフィードバックを送信してください。
2. Create Issue をクリックして、**Create Issue** ページを開きます。
3. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
4. **Create** をクリックします。

第1章 IDENTITY サービスの概要 (KEYSTONE)

クラウドの管理者は、プロジェクト、ユーザー、ロールを管理できます。

プロジェクトとは、リソースの集合が含まれる組織単位のことです。プロジェクト内のロールにユーザーを割り当てることができます。ロールは、特定のプロジェクト内のリソースに対してユーザーが実行できるアクションを定義します。ユーザーは、複数のプロジェクトのロールに割り当てることができます。

各 Red Hat OpenStack (RHOSP) デプロイメントには、プロジェクト内のロールに割り当てられたユーザーが少なくとも 1 人含まれている必要があります。クラウド管理者は、次の操作を実行できます。

- プロジェクトとユーザーを追加、更新、および削除する。
- ユーザーを 1 つまたは複数のロールに割り当て、これらの割り当てを変更または削除する。
- プロジェクトとユーザーを個別に管理する。

Identity サービス (keystone) でユーザー認証を設定して、サービスおよびエンドポイントへのアクセスを制御することも可能です。Identity サービスでは、トークンベースの認証が提供され、LDAP および Active Directory との統合が可能なため、ユーザーとアイデンティティーを外部で管理し、Identity サービスとユーザーデータを同期できます。

1.1. リソース認証情報ファイル

Red Hat OpenStack Platform director をインストールすると、リソース認証情報 (RC) ファイルが自動的に生成されます。

```
# 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"
```

stackrc ファイルを入手して、認証の詳細をシェル環境にエクスポートします。これにより、ローカルの Red Hat OpenStack Platform director API に対してコマンドを実行できます。

オーバークラウドのインストール中に生成される RC ファイルの名前は、デプロイされたスタックの名前に rc という接尾辞が付きます。スタックにカスタム名を指定しない場合、スタックには **overcloud** というラベルが付けられます。作成される RC ファイルの名前は、**overcloudrc** となります。

```
# 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
```

オーバークラウド RC ファイルは、スタックの実際の名前に関係なく、ドキュメントでは **overcloudrc** と呼ばれます。**overcloudrc** ファイルを読み込んで、認証の詳細をシェル環境にエクスポートします。これにより、オーバークラウドクラスターのコントロールプレーン API に対してコマンドを実行できます。自動的に生成された **overcloudrc** ファイルは、**admin** プロジェクトに対して **admin** ユーザーとして認証します。この認証は、プロバイダーネットワークやプロジェクトの作成など、ドメイン管理タスクに役立ちます。

1.2. OPENSTACK リージョン

リージョンは、OpenStack デプロイメントの分割です。各リージョンには、独自の API エンドポイント、ネットワーク、およびコンピュートリソースを含む独自の完全な OpenStack デプロイメントがあります。アクセス制御および Web インターフェイスを提供するために、異なるリージョンが Identity サービス (keystone) および Dashboard サービス (horizon) の 1 つのセットを共有します。Red Hat OpenStack Platform は、単一のリージョンでデプロイされます。デフォルトでは、オーバークラウドのリージョンは、**regionOne** という名前です。Red Hat OpenStack Platform でデフォルトのリージョン名を変更できます。

手順

- **parameter_defaults** の下で、**KeystoneRegion** パラメーターを定義します。

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

- **<sample_region>** を選択した地域名に置き換えます。

注記

オーバークラウドをデプロイした後にリージョン名を変更することはできません。

第2章 ユーザーの管理

クラウド管理者は、Dashboard でユーザーの追加、変更、削除ができます。ユーザーは、1 つまたは複数のプロジェクトに所属することができます。プロジェクトとユーザーは、個別に管理することが可能です。

2.1. DASHBOARD を使用したユーザーの作成

主プロジェクトおよびロールをユーザーに割り当てることができます。OpenStack Dashboard (horizon) を使用して作成するユーザーは、デフォルトで Identity サービスのユーザーです。Identity サービスに含まれる LDAP プロバイダーを設定することで、Active Directory ユーザーを統合できます。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Identity > Users** を選択します。
3. **Create User** をクリックします。
4. ユーザーのユーザー名、メールアドレス、仮のパスワードを入力します。
5. **Primary Project** のリストからプロジェクトを選択します。
6. **Role** の一覧からユーザーのロールを選択します。デフォルトのロールは **member** です。
7. **Create User** をクリックします。

2.2. DASHBOARD を使用したユーザーの編集

主プロジェクトなど、ユーザーの詳細を更新することができます。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Identity > Users** を選択します。
3. **Actions** コラムで、**Edit** をクリックします。
4. **Update User** ウィンドウで、**User Name**、**Email**、**Primary Project** を更新できます。
5. **Update User** をクリックします。

2.3. DASHBOARD を使用したユーザーの有効化または無効化

Dashboard でユーザーを無効にできます。このアクションは、ユーザーの削除とは異なり、元に戻すことができます。

制限

- 一度に複数のユーザーを無効または有効にできません。
- ユーザーの主プロジェクトをアクティブに設定できません。

その結果、無効にしたユーザーは以下を実行できなくなります。

- ダッシュボードにログインする
- RHOSP サービスにアクセスする。
- Dashboard でユーザープロジェクトアクションを実行する。

手順

1. Dashboard に管理ユーザーとしてログインして **Identity > Users** を選択します。
2. **Actions** コラムでドロップダウンリストをクリックし、**Enable User** または **Disable User** を選択します。これにより、**Enabled** コラムの値が **True** または **False** に更新されます。

2.4. DASHBOARD を使用したユーザーの削除

他のユーザーを削除するには、管理者ロールを持つユーザーである必要があります。この操作を元に戻すことはできません。

手順

1. Dashboard に管理ユーザーとしてログインして **Identity > Users** を選択します。
2. 削除するユーザーを選択します。
3. **Delete Users** をクリックします。**Confirm Delete Users** ウィンドウが表示されます。
4. **Delete Users** をクリックしてアクションを確認します。

第3章 ロールの管理

Red Hat OpenStack Platform (RHOSP) はロールベースアクセス制御 (RBAC) のメカニズムを使用して、リソースへのアクセスを管理します。ロールは、ユーザーが実行可能なアクションを定義します。デフォルトでは、事前定義された 2 つのロールがあります。

- プロジェクトにアタッチするメンバーロール
- 管理者以外のユーザーが環境を管理できるようにする管理者ロール

注記

Identity サービス (keystone) には、ロールリストに表示される **reader** ロールも追加されています。**reader** ロールは、Secure RBAC を有効にしている場合にのみ使用してください。

また、環境に固有のカスタムロールを作成することもできます。

3.1. RED HAT OPENSTACK PLATFORM 管理者ロールを理解する

ユーザーに **admin** のロールを割り当てると、このユーザーには、任意のプロジェクトの任意のリソースを表示、変更、作成、または削除する権限があります。このユーザーは、公開されている Glance イメージやプロバイダーネットワークなど、プロジェクト間でアクセスできる共有リソースを作成できます。さらに、**admin** ロールを持つユーザーは、ユーザーを作成または削除し、ロールを管理できます。

ユーザーに **admin** ロールを割り当てるプロジェクトは、**openstack** コマンドが実行されるデフォルトのプロジェクトです。たとえば、**development** という名前のプロジェクトの **admin** ユーザーが次のコマンドを実行すると、**internal-network** という名前のネットワークが **development** プロジェクトに作成されます。

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

admin ユーザーは、**--project** パラメーターを使用して、任意のプロジェクトに **internal-network** を作成できます。

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

3.2. CLI を使用したロールの表示

管理者は、既存のロールの詳細を表示できます。

手順

1. 使用可能な事前定義済みロールをリスト表示します。

```
$ openstack role list
+----------------------------------+----------------+
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
```

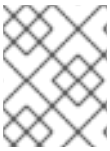
```
| cfea5760d9c948e7b362abc1d06e557f | reader          |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

2. 指定したロールの詳細を表示します。

```
$ openstack role show admin
```

例

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



注記

各ロールに関連付けられた権限に関する情報を取得するには、各 API 呼び出しへのアクセスを監査する必要があります。詳細は、API アクセスの監査 を参照してください。

3.3. CLI を使用したロールの作成と割り当て

管理者は、次の一連のコマンドを使用して、Identity サービス (keystone) クライアントを使用してロールを作成および管理できます。各 Red Hat OpenStack Platform デプロイメントには、最低でもプロジェクト、ユーザー、ロールが 1 つずつあり、それらが連携している必要があります。

複数のプロジェクトにユーザーを割り当てることができます。複数のプロジェクトにユーザーを割り当てるには、ロールを作成して、ユーザーとプロジェクトのペアにそのロールを割り当てます。

注記

ユーザー、ロール、またはプロジェクトの指定には、名前または ID のいずれかを使用できます。

手順

1. **new-role** ロールを作成します。

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

2. プロジェクトにユーザーを割り当てるには、まず以下のコマンドを使用して、ユーザー、ロール、およびプロジェクト名または ID を見つけます。

- openstack user list
- openstack role list
- openstack project list

3. ユーザーとプロジェクトのペアにロールを割り当てます。

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

以下の例では、**demo** プロジェクトの **admin** ユーザーに **admin** ロールを割り当てています。

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

4. **admin** ユーザーのロール割り当てを確認します。

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

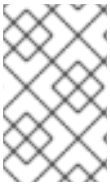
以下の例では、**admin** ユーザーが **admin** ロールで **demo** プロジェクトに割り当てられていることを確認します。

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

3.4. 暗黙的なロールの作成

Identity サービス (keystone) は、ユーザーが特定のロールに割り当てられていることを確認してアクセス制御を適用します。Identity サービスは暗黙的なロール割り当てを使用します。ユーザーをロールに明示的に割り当てると、ユーザーは追加のロールにも暗黙的に割り当てられます。Red Hat OpenStack Platform での、デフォルトの暗黙的なロールを表示することができます。

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



注記

Identity サービス (keystone) には、ロールリストに表示される **reader** ロールも追加されています。**reader** ロールは、Secure RBAC を有効にしている場合にのみ使用してください。

より高い権限を持つロールには、より低い権限を持つロールに関連付けられた権限が含まれます。上記のデフォルトの暗黙的なロールでは、admin には member が、member には reader が含まれます。暗黙的なロールを使用すると、ユーザーのロール割り当てが累積的に処理されるため、ユーザーは下位ロールを継承します。

カスタムロールを使用する場合は、暗黙的な関連付けを作成できます。

注記

新たなロールを作成する場合、デフォルトではこのロールには **member** ロールと同じアクセスポリシーが設定されます。カスタムロールの一意のポリシーの作成については、アクセス制御にポリシーファイルを使用する を参照してください。

手順

- 次のコマンドを使用して、別のロールに含まれるロールを指定します。

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

検証

- すべての暗黙的なロールをリスト表示します。

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

暗黙的な関連付けに誤りがある場合は、変更を元に戻すことができます。

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

第4章 グループの管理

Identity サービス (keystone) グループを使用すると、一定のパーミッションを複数のユーザーアカウントに割り当てることができます。

4.1. CLI を使用してグループを設定する

グループを作成し、グループに権限を割り当てます。グループのメンバーは、グループに割り当てたものと同じ権限を継承します。

1. **grp-Auditors** というグループを作成します。

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

2. keystone グループのリストを表示します。

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

3. **member** ロールを使用して **demo** プロジェクトにアクセスするための **grp-Auditors** グループパーミッションを付与します。

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

4. 既存のユーザー **user1** を **grp-Auditors** グループに追加します。

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

5. **user1** が **grp-Auditors** のメンバーであることを確認します。

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

6. **user1** に割り当てられている有効なパーミッションを確認します。

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

```
+----------------------------------+----------------------------------+-------+-----------------------------
--+--------+-----------+
| 9fe2ff9ee4384b1894a90878d3e92bab | 3fefe5b4f6c948e6959d1feaef4822f2 |       |
0ce36252e2fb4ea8983bed2a568fa832 |        | False     |
+----------------------------------+----------------------------------+-------+-----------------------------
--+--------+-----------+
```

4.2. ダッシュボードを使用してグループを設定する

Dashboard を使用して keystone グループのメンバーシップを管理できます。ただし、グループにロール権限を割り当てるには、コマンドラインを使用する必要があります。詳細は、CLI でグループの設定を参照してください。

4.2.1. グループの作成

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Groups** を選択します。
3. **+Create Group** をクリックします。
4. グループの名前と説明を入力します。
5. **Create Group** をクリックします。

4.2.2. グループメンバーシップの管理

Dashboard を使用して keystone グループのメンバーシップを管理できます。

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Groups** を選択します。
3. 編集するグループの **Manage Members** をクリックします。
4. **Add users** を使用して、グループにユーザーを追加します。ユーザーを削除する場合には、そのユーザーのチェックボックスを選択して、**Remove users** をクリックします。

第5章 クォータ管理

クラウド管理者は、プロジェクトのクォータを設定、管理できます。各プロジェクトには、リソースが割り当てられており、プロジェクトユーザーには、これらのリソースを使用するパーミッションが付与されます。これにより、相互のパーミッションやリソースを干渉することなく、複数のプロジェクトが単一のクラウドを使用できます。リソースクォータのセットは、新規プロジェクトの作成時に事前設定されます。クォータには、プロジェクトに割り当て可能な仮想 CPU、インスタンス、RAM、および Floating IP の数量が含まれます。クォータは、プロジェクトレベルと、プロジェクトのユーザーレベルの両方で実行できます。Dashboard を使用して、新規/既存プロジェクトの Compute または Block Storage のクォータを設定または変更できます。詳細は、プロジェクトの管理 を参照してください。

5.1. ユーザーのコンピュートクォータの表示

ユーザーに現在設定されているクォータの値をリスト表示するには、以下のコマンドを実行します。

手順

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

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. ユーザーのコンピュートクォータの更新

特定のクォータ値を更新するには、以下のコマンドを実行します。

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

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

注記

quota-update コマンドのオプションリストを表示するには、以下を実行します。

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

5.3. ユーザーの OBJECT STORAGE クォータの設定

オブジェクトストレージクォータは、以下のカテゴリーに分類できます。

- コンテナークォータ: 合計サイズ (バイト単位) または単一のコンテナーで保存可能なオブジェクト数を制限します。
- アカウントクォータ: Object Storage サービスでユーザーが利用可能な合計サイズ (バイト単位) を制限します。

コンテナークォータまたはアカウントクォータのいずれかを設定するには、Object Storage プロキシーサーバーにおいて、**proxy-server.conf** ファイルの **[pipeline:main]** セクションに **container_quotas** または **account_quotas** (または両方) のパラメーターを追加する必要があります。

```
[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
```

オブジェクトストレージクォータの表示および更新には、以下のコマンドを使用します。プロジェクトに含まれるすべてのユーザーには、そのプロジェクトに指定されているクォータが表示されます。プロジェクトに設定されているオブジェクトストレージのクォータを更新するには、そのプロジェクトの ResellerAdmin のロールが必要です。

アカウントクォータを表示するには、以下のコマンドを実行します。

```
# 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
```

クォータを更新するには、以下を実行します。

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

たとえば、アカウントに 5 GB のクォータを指定します。

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

第6章 プロジェクトの管理

クラウド管理者は、プロジェクトを作成、管理できます。プロジェクトは共有仮想リソースのプールで、OpenStack のユーザーおよびグループを割り当てることができます。各プロジェクトで共有仮想リソースのクォータを設定できます。Red Hat OpenStack Platform では、相互に権限およびリソースが干渉しない複数のプロジェクトを作成できます。ユーザーは、複数のプロジェクトに割り当てることができます。各ユーザーには、割り当てられた各プロジェクトに指定されたロールが設定されている必要があります。

6.1. プロジェクトの作成

プロジェクトを作成し、プロジェクトにメンバーを追加し、プロジェクトリソースの制限を設定します。

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Projects** を選択します。
3. **Create Project** をクリックします。
4. **Project Information** タブで、プロジェクトの名前と説明を入力します。デフォルトで、**Enabled** のチェックボックスが選択されています。
5. プロジェクトへのメンバーの追加は、**Project Members** タブの **All Users** リストから行います。
6. **Quotas** タブで、プロジェクトのリソースの上限を指定します。
7. **Create Project** をクリックします。

6.2. プロジェクトの編集

プロジェクトを編集して名前や説明を変更したり、プロジェクトを有効化または一時的に無効化したり、プロジェクトのメンバーを更新したりすることができます。

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Projects** を選択します。
3. プロジェクトの **Actions** コラムで、下向きの三角をクリックして **Edit Project** をクリックします。
4. **Edit Project** ウィンドウでプロジェクトを更新して名前や説明を変更したり、プロジェクトを有効化または一時的に無効化したりすることができます。
5. **Project Members** タブで、必要に応じてメンバーをプロジェクトに追加または削除します。
6. **Save** をクリックします。

注記

デフォルトで、**Enabled** のチェックボックスが選択されています。プロジェクトを一時的に無効にするには、**Enabled** のチェックボックスのチェックマークを外します。無効なプロジェクトを有効にするには、**Enabled** チェックボックスを選択します。

6.3. プロジェクトの削除

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Projects** を選択します。
3. 削除するプロジェクトを選択します。
4. **Delete Projects** をクリックします。**Confirm Delete Projects** ウィンドウが表示されます。
5. **Delete Projects** をクリックしてアクションを確認します。

プロジェクトが削除され、ユーザーとのペアリングの関連付けは解除されます。

6.4. プロジェクトクォータの更新

クォータとは、クラウドリソースを最適化するためにプロジェクトごとに設定する操作の制約のことです。クォータを設定して、通知なしにプロジェクトのリソースが使い果たされないようにします。クォータは、プロジェクトレベルとプロジェクトのユーザーレベルの両方で適用できます。

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Projects** を選択します。
3. プロジェクトの **Actions** コラムで、下向きの三角をクリックして **Modify Quotas** をクリックします。
4. **Quota** タブで、必要に応じてプロジェクトクォータを変更します。
5. **Save** をクリックします。

注記

現時点では、**nested quotas** はまだサポートされていません。そのため、クォータはプロジェクトとサブプロジェクトで別々に管理する必要があります。

6.5. アクティブなプロジェクトの変更

Dashboard を使用してプロジェクトのオブジェクトを操作できるように、プロジェクトをアクティブなプロジェクトとして設定します。プロジェクトをアクティブなプロジェクトとして設定するには、プロジェクトのメンバーである必要があります。また、**Set as Active Project** オプションを有効にするには、ユーザーが複数のプロジェクトのメンバーである必要があります。プロジェクトを再度有効にしない限り、無効なプロジェクトをアクティブなプロジェクトとして設定することはできません。

1. 管理者権限を持つユーザーとして Dashboard にログインします。
2. **Identity > Projects** を選択します。
3. プロジェクトの **Actions** コラムで、下向きの三角をクリックして **Set as Active Project** をクリックします。
4. または、管理者権限のないユーザーで、プロジェクトの **Actions** コラムの下向きの三角をクリックして **Set as Active Project** をクリックすると、このコラムのデフォルトアクションになります。

6.6. プロジェクトの階層

Identity サービス (keystone) のマルチテナンシーを使用して、プロジェクトを入れ子状にできます。マルチテナンシーにより、サブプロジェクトは親プロジェクトのロール割り当てを継承できます。

6.6.1. 階層化されたプロジェクトとサブプロジェクトの作成

階層型マルチテナンシー (HMT) は keystone のドメインとプロジェクトを使用して実装できます。まず最初に新規ドメインを作成して、そのドメイン内にプロジェクトを作成します。これで、そのプロジェクトにサブプロジェクトを追加できるようになります。また、ユーザーをサブプロジェクトの **admin** ロールに追加すると、そのサブプロジェクトの管理者に昇格できます。

注記

keystone の使用する HMT の構造は、現在 Dashboard では表示されません。

手順

1. **corp** という名前の keystone ドメインを新規作成します。

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

2. **corp** ドメイン内に親プロジェクト (**private-cloud**) を作成します。

```
$ 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. **private-cloud** の親プロジェクト内で **corp** ドメインも指定して、サブプロジェクト (**dev**) を作成します。

```
$ 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. **qa** という名前のサブプロジェクトをもう 1 つ作成します。

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

注記

Identity API を使用してプロジェクトの階層を確認することができます。詳しくは、https://developer.openstack.org/api-ref/identity/v3/index.html?expanded=show-project-details-detail を参照してください。

6.6.2. 階層構造のプロジェクトへのアクセスの設定

デフォルトでは、新規作成したプロジェクトにはロールは割り当てられません。親プロジェクトに対するロールのパーミッションを割り当てるときに、**--inherited** フラグを指定して、サブプロジェクトが親プロジェクトからパーミッションを継承するように指定できます。たとえば、親プロジェクトに対する **admin** ロールのアクセス権のあるユーザーには、サブプロジェクトへの **admin** アクセス権も付与されます。

ユーザーへのアクセス権の付与

1. プロジェクトに割り当てられている既存のパーミッションを確認します。

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

2. 既存のロールを確認します。

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

```
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

3. ユーザーアカウント **user1** に、**private-cloud** プロジェクトに対するアクセス権を付与します。

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

このコマンドを **--inherited** フラグを使用して再実行します。その結果、**user1** には **private-cloud** のサブプロジェクト (ロールの割り当てが継承されている) に対するアクセス権も付与されます。

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

4. パーミッションの更新の結果を確認します。

```
$ 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 ユーザーは、**qa** プロジェクトと **dev** プロジェクトへのアクセス権を継承しています。さらに、**--inherited** フラグが親プロジェクトに適用されたため、**user1** は後から作成されたサブプロジェクトにもアクセスできるようになりました。

ユーザーからのアクセス削除

明示的に割り当てられたパーミッションと継承されたパーミッションは別々に削除する必要があります。

1. 明示的に割り当てられたロールからユーザーを削除します。

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

2. 変更の結果を確認します。継承されたパーミッションがまだ存在している点に注意してください。

```
$ 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. 継承されたパーミッションを削除します。

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

4. 変更の結果を確認します。継承されたパーミッションが削除され、出力が空になりました。

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

6.6.3. Reseller プロジェクトの概要

Reseller プロジェクトでは、複数のドメインを階層化することを目標としています。このようなドメインでは、1 つのサブドメインは、完全に有効化された 1 つのクラウドを表現し、最終的にはクラウドの部分的な再販を考慮することができます。この開発の作業は複数の段階に分かれています。第 1 段階については以下に説明します。

Reseller の第 1 段階

Reseller (第 1 段階) は、階層化されたプロジェクトとサブプロジェクトの作成 に記載されている階層型マルチテナンシー (HMT) の延長です。従来 keystone ドメインは、データベースバックエンド内に独自のテーブルを備えた、ユーザーとプロジェクトを保管するためのコンテナーとすることを目的としていました。その結果、ドメインは独自のテーブルには保管されなくなり、プロジェクトのテーブルにマージされました。

- ドメインは、ひとつのプロジェクトタイプとなり、**is_domain** フラグで区別されます。
- ドメインは、プロジェクト階層の最上位のプロジェクトを表します。ドメインは、プロジェクト階層のルートです。
- **projects** サブパスを使用してドメインの作成と取得を行うように API が更新されました。
  - 新規ドメインを作成するには、**is_domain** フラグを true に指定してプロジェクトを作成します。
  - ドメインであるプロジェクトをリスト表示します。**is_domain** クエリーパラメーターを含むプロジェクトを取得します。

6.7. プロジェクトのセキュリティー管理

セキュリティーグループとは、プロジェクトのインスタンスに割り当て可能な IP フィルターのルールセットで、インスタンスへのネットワークのアクセス権限を定義します。セキュリティーグループはプロジェクト別になっており、プロジェクトメンバーは自分のセキュリティーグループのデフォルトルールを編集して新規ルールセットを追加できます。

すべてのプロジェクトにはデフォルトのセキュリティーグループが存在し、他にセキュリティーグループが定義されていないインスタンスに対して適用されます。このセキュリティーグループは、デフォルト値を変更しない限り、インスタンスへの受信トラフィックをすべて拒否し、送信トラフィックのみを

許可します。

セキュリティーグループは、インスタンス作成時に直接インスタンスに適用するか、実行中のインスタンスのポートに適用できます。

注記

インスタンスの作成中に、ロールベースのアクセス制御 (RBAC) 共有セキュリティーグループをインスタンスに直接適用することはできません。RBAC 共有セキュリティーグループをインスタンスに適用するには、最初にポートを作成し、共有セキュリティーグループをそのポートに適用してから、そのポートをインスタンスに割り当てる必要があります。セキュリティーグループのポートへの追加 を参照してください。

必要な出力を許可するグループを作成せずに、デフォルトのセキュリティーグループを削除しないでください。たとえば、インスタンスが DHCP とメタデータを使用している場合、インスタンスには、DHCP サーバーとメタデータエージェントへの出力を許可するセキュリティーグループルールが必要です。

6.7.1. セキュリティーグループの作成

セキュリティーグループを作成して、セキュリティールールを設定できるようにします。たとえば、ICMP トラフィックを有効にしたり、HTTP リクエストを無効にしたりできます。

手順

1. Dashboard で **Project** > **Compute** > **Access & Security**を選択します。
2. **Security Groups** タブで、**Create Security Group** をクリックします。
3. グループの名前と説明を入力して、**Create Security Group** をクリックします。

6.7.2. セキュリティーグループルールの追加

デフォルトでは、新しいグループには、送信アクセスのルールのみが指定されます。他のアクセスを指定するには、新しいルールを追加する必要があります。

手順

1. Dashboard で **Project** > **Compute** > **Access & Security**を選択します。
2. **Security Groups** タブで、編集するセキュリティーグループの **Manage Rules** をクリックします。
3. **Add Rule** をクリックして、新規ルールを追加します。
4. ルールの値を指定して、**Add** をクリックします。
   以下のルールのフィールドは必須です。

   ルール

   ルールタイプ。ルールテンプレート (例: 'SSH') を指定する場合には、そのフィールドは自動的に入力されます。

   - TCP: 一般的には、システム間のデータの交換や、エンドユーザーの通信に使用されます。

- UDP: 一般的には、システム間のデータ交換に (特にアプリケーションレベルで) 使用されます。
- ICMP: 一般的には、ルーターなどのネットワークデバイスがエラーや監視メッセージを送信するのに使用されます。

方向

受信 (インバウンド) または送信 (アウトバウンド)

開放するポート

TCP または UDP ルールでは、開放する **Port** または **Port Range** (単一のポートまたはポートの範囲) を入力します。

- ポート範囲では、**From Port** と **To Port** フィールドにポートの値を入力します。
- 単一のポートの場合は **Port** フィールドにポートの値を入力します。

型

ICMP ルールのタイプ。'-1:255' の範囲で指定する必要があります。

コード

ICMP ルールのコード。'-1:255' の範囲で指定する必要があります。

接続相手

このルールが適用されるトラフィックの接続元

- CIDR (Classless Inter-Domain Routing): 指定のブロック内の IP へのアクセスを制限する IP アドレスブロック。ソースフィールドに CIDR を入力します。
- セキュリティーグループ: グループ内のインスタンスが他のグループインスタンスにアクセスできるようにするソースのセキュリティーグループ。

6.7.3. セキュリティーグループルールの削除

不要になったセキュリティーグループルールを削除します。

手順

1. Dashboard で **Project > Compute > Access & Security**を選択します。
2. **Security Groups** タブで、セキュリティーグループの **Manage Rules** をクリックします。
3. セキュリティーグループルールを選択し、**Delete Rule** ボタンをクリックします。
4. 再度、**Delete Rule** をクリックします。

注記

削除の操作は元に戻せません。

6.7.4. セキュリティーグループの削除

不要になったセキュリティーグループを削除します。

手順

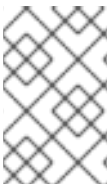
1. Dashboard で **Project** > **Compute** > **Access & Security**を選択します。
2. **Security Groups** タブで、グループを選択して、**Delete Security Groups** をクリックします。
3. **Delete Security Groups** をクリックします。

注記

削除の操作は元に戻せません。

第7章 ドメインの管理

Identity サービス (keystone) ドメインは、keystone で作成可能な追加の名前空間です。keystone ドメインを使用して、ユーザー、グループ、プロジェクトを分割します。異なる LDAP または Active Directory 環境でユーザーを認証するように、これらの分離されたドメインを設定することも可能です。詳細は、Identity サービスとの統合 を参照してください。



注記

Identity サービスには、**Default** という名前のドメインが組み込まれています。このドメインは、サービスアカウント専用に確保し、ユーザーアカウント用には別のドメインを作成することを推奨します。

7.1. ドメインリストの表示

openstack domain list コマンドでドメインの一覧を表示できます。

```
$ 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. 新規ドメインの作成

openstack domain create コマンドで新しいドメインを作成できます。

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

7.3. ドメインの詳細表示

openstack domain show コマンドで、ドメインの詳細を表示できます。

```
$ openstack domain show TestDomain
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
```

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

7.4. ドメインの無効化

必要に応じてドメインを無効化または有効化できます。

手順

1. **--disable** オプションでドメインを無効にします。

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

2. ドメインが無効化されたことを確認します。

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

3. 必要に応じて **--enable** オプションを使用して、ドメインを再度有効にします。

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

第8章 アプリケーション認証情報

アプリケーション認証情報 を使用することで、設定ファイルにユーザーアカウントの認証情報を埋め込むことを回避できます。代わりに、ユーザーは、1つのプロジェクトへのアクセスを委譲された、個別のシークレットを持つアプリケーション認証情報を作成します。ユーザーは、委譲されたアクセス権限をそのプロジェクト内の単一のロールに制限することもできます。これにより、すべてのプロジェクトおよびロールではなく、サービスが機能するのに必要な1つのプロジェクトおよびロールへのアクセス権限のみ付与でき、最小権限の原則に準拠できます。

この手法を使用すると、ユーザー認証情報を公開せずに API を消費することが可能になり、アプリケーションは埋め込まれたユーザー認証情報を必要とせずに Keystone に対して認証することができます。

アプリケーション認証情報を使用してトークンを生成し、アプリケーションの **keystone_authtoken** 設定を定義できます。これらのユースケースは、これ以降のセクションで説明します。

注記

アプリケーション認証情報は、その認証情報を作成したユーザーアカウントに従属します。したがって、そのアカウントが削除されたり、該当するロールにアクセスできなくなったりすると、アプリケーション認証情報は機能しなくなります。

8.1. アプリケーション認証情報を使用したトークンの生成

ユーザーは、Dashboard のセルフサービス機能として、アプリケーション認証情報を利用できます。以下の例は、ユーザーがアプリケーション認証情報を作成し、それを使用してトークンを生成する方法を示しています。

1. テスト用プロジェクトおよびユーザーアカウントを作成します。
   a. **AppCreds** という名前のプロジェクトを作成します。

   ```
   $ openstack project create AppCreds
   ```

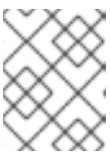
   b. **AppCredsUser** という名前のユーザーを作成します。

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

   c. **AppCredsUser** に、**AppCreds** プロジェクトの **member** ロールへのアクセス権限を付与します。

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

2. **AppCredsUser** として Dashboard にログインし、アプリケーション認証情報を作成します。**Overview → Identity → Application Credentials → +Create Application Credential** に移動します。

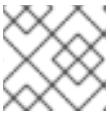


注記

Your Application Credential ポップアップウィンドウを閉じると再アクセスできなくなるため、**clouds.yaml** ファイルの内容をダウンロードしてください。

3. CLI を使用して **/home/stack/.config/openstack/clouds.yaml** という名前のファイルを作成し、**clouds.yaml** ファイルの内容を貼り付けます。

```
# 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"
```



注記

これらの値は、実際のデプロイメントとは異なる場合があります。

4. アプリケーション認証情報を使用してトークンを生成します。以下のコマンドを使用する場合、特定のユーザーとしてソースを提供しないでください。また、**clouds.yaml** ファイルのディレクトリーを現在の作業ディレクトリーにする必要があります。

```
[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_P9CbhuoIt |
| project_id | 1a74eabbf05c41baadd716179bb9e1da
|
| user_id    | ef679eeddfd14f8b86becfd7e1dc84f2
|
+------------+-----------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------+
```

注記

init() got an unexpected keyword argument 'application_credential_secret' のようなエラーメッセージが表示される場合は、まだ以前の認証情報にソースを提供している可能性があります。新しい環境の場合は、**sudo su - stack** を実行します。

8.2. アプリケーション認証情報とアプリケーションの統合

アプリケーション認証情報を使用して、アプリケーションを keystone に対して認証できます。アプリケーション認証情報を使用する場合 **keystone_authtoken** の設定では、認証タイプ **v3applicationcredential** が使用され、認証情報の作成プロセスで受け取った認証情報が含まれます。以下の値を設定します。

- **application_credential_secret**: アプリケーション認証情報のシークレット
- **application_credential_id**: アプリケーション認証情報の ID
- (オプション) **application_credential_name**: ID ではなく、名前付きのアプリケーション認証情報を使用する場合に、このパラメーターを使用することがあります。

以下に例を示します。

```
[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. アプリケーション認証情報の管理

コマンドラインを使用して、アプリケーション認証情報を作成および削除できます。

create サブコマンドにより、現在ソースを提供しているアカウントに基づいてアプリケーション認証情報が作成されます。たとえば、**admin** ユーザーとしてソースを提供している場合に認証情報を作成すると、同じロールがアプリケーション認証情報に付与されます。

```
$ 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       | fVnqa6l_XeRDDkmQnB5lx361W1jHtOtw3ci_mf_tOID-09MrPAzkU7mv-by8ykEhEa1QLPFJLNV4cS2Roo9lOg |
| unrestricted | False                                                                                   |
+--------------+-----------------------------------------------------------------------------------------+
```

警告

--unrestricted パラメーターを使用すると、アプリケーション認証情報で他のアプリケーション認証情報と信頼を作成および削除できます。これは潜在的に危険な動作であり、デフォルトでは無効になっています。**--unrestricted** パラメーターは、他のアクセスルールと組み合わせて使用できません。

デフォルトでは、付与されるロールのメンバーシップには、認証情報を作成したアカウントに割り当てられたすべてのロールが含まれます。特定ロールだけを対象にアクセスを委譲して、ロールのメンバーシップを限定することができます。

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

アプリケーション認証情報を削除するには、以下のコマンドを実行します。

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

8.4. アプリケーション認証情報の交換

アプリケーション認証情報は、その認証情報を作成したユーザーアカウントにバインドされます。したがって、そのユーザーアカウントが削除されたり、ユーザーが委譲されたロールにアクセスできなくなったりすると、アプリケーション認証情報は無効になります。そのため、必要に応じて新しいアプリケーション認証情報を生成する準備が必要になります。

設定ファイル用の既存のアプリケーション認証情報の置き換え

(設定ファイルを使用して) アプリケーションに割り当てられたアプリケーション認証情報を更新します。

1. 新しいアプリケーション認証情報のセットを作成します。
2. アプリケーションの設定ファイルに新しい認証情報を追加し、既存の認証情報と置き換えます。詳細は、アプリケーション認証情報とアプリケーションの統合 を参照してください。
3. アプリケーションのサービスを再起動して、変更を適用します。

4. 必要に応じて、古いアプリケーション認証情報を削除します。コマンドラインオプションの詳細は、アプリケーション認証情報の管理 を参照してください。

clouds.yaml 内の既存のアプリケーション認証情報の置き換え

clouds.yaml で使用されるアプリケーション認証情報を置き換える場合は、OpenStack ユーザー認証情報を使用して置換認証情報を作成する必要があります。デフォルトでは、アプリケーション認証情報を使用して別のアプリケーション認証情報のセットを作成できません。**openstack application credential create** コマンドは、現在ソースを提供しているアカウントに基づいてアプリケーション認証情報を作成します。

1. 有効期限が近づいている認証認証情報を最初に作成した OpenStack ユーザーとして認証します。たとえば、アプリケーション認証情報を使用してトークンを生成する 手順を使用した場合は、**AppCredsUser** として再度ログインする必要があります。

2. **AppCred2** という名前のアプリケーション認証情報を作成します。これは、OpenStack Dashboard または **openstack** CLI インターフェイスを使用して実行できます。

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

3. 直前のコマンドの出力から **id** パラメーターおよび **secret** パラメーターをコピーします。**secret** パラメーター値に再度アクセスできません。

4. **${HOME}/.config/openstack/clouds.yaml** ファイル内の **application_credential_id** パラメーター値と **application_credential_secret** パラメーター値を、コピーした **Secret** 値と **ID** 値に置き換えます。

検証

1. clouds.yaml でトークンを生成し、認証情報が予想通りに機能していることを確認します。以下のコマンドを使用する場合、特定のユーザーとしてソースを提供しないでください。また、**clouds.yaml** ファイルのディレクトリーを現在の作業ディレクトリーにする必要があります。

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

出力例:

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

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