Red Hat OpenStack Platform 17.1

OpenStack ID 리소스 관리

사용자 및 keystone 인증 구성

Last Updated: 2024-11-22

Red Hat OpenStack Platform 17.1 OpenStack ID 리소스 관리

사용자 및 keystone 인증 구성

OpenStack Team
rhos-docs@redhat.com

법적 공지

초록

애플리케이션 자격 증명, 사용자, 역할, 프로젝트, 할당량을 관리합니다.

차례

머리말 3

보다 포괄적 수용을 위한 오픈 소스 용어 교체 4

RED HAT 문서에 관한 피드백 제공 5

1장. IDENTITY 서비스(KEYSTONE) 소개 6
1.1. 리소스 인증 정보 파일 6
1.2. OPENSTACK 리전 7

2장. 사용자 관리 8
2.1. 대시보드를 사용하여 사용자 생성 8
2.2. 대시보드로 사용자 편집 8
2.3. 대시보드를 사용하여 사용자 활성화 또는 비활성화 8
2.4. 대시보드를 사용하여 사용자 삭제 9

3장. 역할 관리 10
3.1. RED HAT OPENSTACK PLATFORM 관리자 역할 이해 10
3.2. CLI를 사용하여 역할 보기 10
3.3. CLI를 사용하여 역할 생성 및 할당 11
3.4. 암시적 역할 생성 12

4장. 그룹 관리 14
4.1. CLI를 사용하여 그룹 구성 14
4.2. 대시보드를 사용하여 그룹 구성 15

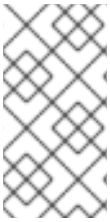
5장. 할당량 관리 16
5.1. 사용자의 COMPUTE 할당량 보기 16
5.2. 사용자의 컴퓨팅 할당량 업데이트 16
5.3. 사용자의 오브젝트 스토리지 할당량 설정 17

6장. 프로젝트 관리 19
6.1. 프로젝트 생성 19
6.2. 프로젝트 편집 19
6.3. 프로젝트 삭제 19
6.4. 프로젝트 할당량 업데이트 20
6.5. 활성 프로젝트 변경 20
6.6. 프로젝트 계층 구조 20
6.7. 프로젝트 보안 관리 24

7장. 도메인 관리 27
7.1. 도메인 목록 보기 27
7.2. 새 도메인 생성 27
7.3. 도메인의 세부 정보 보기 27
7.4. 도메인 비활성화 28

8장. 애플리케이션 인증 정보 29
8.1. 애플리케이션 인증 정보를 사용하여 토큰 생성 29
8.2. 애플리케이션과 애플리케이션 인증 정보 통합 30
8.3. 애플리케이션 인증 정보 관리 31
8.4. 애플리케이션 인증 정보 교체 32

머리말



참고

인스턴스를 생성하는 동안 RBAC(역할 기반 액세스 제어) 공유 보안 그룹을 인스턴스에 직접 적용할 수 없습니다. 인스턴스에 RBAC-공유 보안 그룹을 적용하려면 먼저 포트를 만들고, 공유 보안 그룹을 해당 포트에 적용한 다음 해당 포트를 인스턴스에 할당해야 합니다. 포트에 보안 그룹 추가를 참조하십시오.

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

Jira에서 문서 피드백 제공

Create Issue 양식을 사용하여 OpenShift (RHOSO) 또는 이전 Red Hat OpenStack Platform (RHOSP)의 Red Hat OpenStack Services 문서에 대한 피드백을 제공합니다. RHOSO 또는 RHOSP 문서에 대한 문제를 생성할 때 RHOSO Jira 프로젝트에 문제가 기록되어 피드백의 진행 상황을 추적할 수 있습니다.

문제 생성 양식을 완료하려면 Jira에 로그인해야 합니다. Red Hat Jira 계정이 없는 경우 https://issues.redhat.com 에서 계정을 생성할 수 있습니다.

1. 다음 링크를 클릭하여 **문제 생성** 페이지를 엽니다.
   https://issues.redhat.com/secure/CreateIssueDetails!init.jspa?pid=12336920&summary=Documentation%20feedback:%20%3CAdd%20summary%20here%3E&i
   <Include+the+documentation+URL,+the%20chapter+or+section+number,+and+a+detailed+descrip

2. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.

3. **생성**을 클릭합니다.

1장. IDENTITY 서비스(KEYSTONE) 소개

클라우드 관리자는 프로젝트, 사용자 및 역할을 관리할 수 있습니다.

프로젝트는 리소스 컬렉션을 포함하는 조직 단위입니다. 프로젝트 내의 역할에 사용자를 할당할 수 있습니다. 역할은 해당 사용자가 지정된 프로젝트 내의 리소스에서 수행할 수 있는 작업을 정의합니다. 사용자는 여러 프로젝트에서 역할을 할당할 수 있습니다.

각 RHOSP(Red Hat OpenStack) 배포에는 프로젝트 역할에 할당된 사용자가 하나 이상 포함되어야 합니다. 클라우드 관리자는 다음을 수행할 수 있습니다.

- 프로젝트 및 사용자를 추가, 업데이트 및 삭제합니다.
- 사용자를 하나 이상의 역할에 할당하고 이러한 할당을 변경하거나 제거합니다.
- 프로젝트와 사용자를 서로 독립적으로 관리합니다.

ID 서비스(keystone)를 사용하여 사용자 인증을 구성하여 서비스 및 엔드포인트에 대한 액세스를 제어할 수도 있습니다. ID 서비스는 토큰 기반 인증을 제공하고 LDAP 및 Active Directory와 통합할 수 있으므로 외부 사용자 및 ID를 관리하고 사용자 데이터를 ID 서비스와 동기화할 수 있습니다.

1.1. 리소스 인증 정보 파일

Red Hat OpenStack Platform director를 설치하면 리소스 인증 정보(RC) 파일이 자동으로 생성됩니다.

```
# 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 파일을 소싱하여 인증 세부 정보를 쉘 환경으로 내보냅니다. 이를 통해 로컬 Red Hat OpenStack Platform director API에 대해 명령을 실행할 수 있습니다.

오버클라우드 설치 중에 생성된 RC 파일의 이름은 'rc' 접미사가 배포된 스택의 이름입니다. 스택에 사용자 지정 이름을 지정하지 않으면 스택에 **overcloud** 레이블이 지정됩니다. RC 파일은 **overcloudrc** 라는 이름으로 생성됩니다.

```
# 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
```

오버클라우드 RC 파일은 스택의 실제 이름에 관계없이 문서에서 **overcloudrc** 라고 합니다. **overcloudrc** 파일을 가져와 인증 세부 정보를 쉘 환경으로 내보냅니다. 이를 통해 오버클라우드 클러스터의 컨트롤 플레인 API에 대해 명령을 실행할 수 있습니다. 자동으로 생성된 **overcloudrc** 파일은 **admin** 사용자로 **admin** 프로젝트에 대해 인증합니다. 이 인증은 공급자 네트워크 또는 프로젝트 생성과 같은 도메인 관리 작업에 중요합니다.

1.2. OPENSTACK 리전

리전은 OpenStack 배포의 분할입니다. 각 리전에는 자체 API 엔드포인트, 네트워크 및 컴퓨팅 리소스를 포함하여 자체 OpenStack 배포가 있습니다. 다른 리전은 하나의 ID 서비스(keystone) 및 대시보드 서비스(horizon) 서비스를 공유하여 액세스 제어 및 웹 인터페이스를 제공합니다. Red Hat OpenStack Platform은 단일 리전과 함께 배포됩니다. 기본적으로 오버클라우드 리전의 이름은 **regionOne** 입니다. Red Hat OpenStack Platform에서 기본 리전 이름을 변경할 수 있습니다.

프로세스

- **parameter_defaults** 에서 **KeystoneRegion** 매개변수를 정의합니다.

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

- & **lt;sample_region** >을 선택한 리전 이름으로 바꿉니다.

참고

오버클라우드를 배포한 후에는 리전 이름을 수정할 수 없습니다.

2장. 사용자 관리

클라우드 관리자는 대시보드에서 사용자를 추가, 수정, 삭제할 수 있습니다. 사용자는 하나 이상의 프로젝트의 멤버일 수 있습니다. 프로젝트와 사용자를 서로 독립적으로 관리할 수 있습니다.

2.1. 대시보드를 사용하여 사용자 생성

사용자에게 기본 프로젝트 및 역할을 할당할 수 있습니다. OpenStack Dashboard(horizon)를 사용하여 생성하는 사용자는 기본적으로 ID 서비스 사용자입니다. ID 서비스에 포함된 LDAP 공급자를 구성하여 Active Directory 사용자를 통합할 수 있습니다.

절차

1. admin 사용자로 대시보드에 로그인합니다.
2. **Identity > Users**를 선택합니다.
3. **사용자 생성을** 클릭합니다.
4. 사용자의 사용자 이름, 이메일, 사전 암호를 입력합니다.
5. **기본 프로젝트** 목록에서 프로젝트를 선택합니다.
6. **Role** 목록에서 사용자의 역할을 선택합니다. 기본 역할은 **member** 입니다.
7. **사용자 생성을** 클릭합니다.

2.2. 대시보드로 사용자 편집

기본 프로젝트를 포함하여 사용자 세부 정보를 업데이트할 수 있습니다.

절차

1. 대시보드에 admin 사용자로 로그인합니다.
2. **Identity > Users**를 선택합니다.
3. **작업** 열에서 **편집** 을 클릭합니다.
4. **사용자 업데이트 창에서 사용자 이름,이메일**, **기본 프로젝트**를 업데이트할 수 있습니다.
5. **사용자 업데이트**를 클릭합니다.

2.3. 대시보드를 사용하여 사용자 활성화 또는 비활성화

대시보드를 사용하여 사용자를 비활성화할 수 있습니다. 이 작업은 사용자를 삭제하는 것과 달리 되돌릴 수 있습니다.

제한 사항:

- 한 번에 두 명 이상의 사용자를 비활성화하거나 활성화할 수 없습니다.
- 사용자의 기본 프로젝트를 active로 설정할 수 없습니다.

결과적으로 비활성화한 사용자는 다음을 수행할 수 없습니다.

- 대시보드에 로그인합니다.
- RHOSP 서비스에 액세스합니다.
- 대시보드에서 모든 user-project 작업을 수행합니다.

절차

1. 대시보드에서 관리자로서 **ID > 사용자를 선택합니다**.
2. **작업** 열에서 화살표를 클릭하고 **사용자 활성화 또는사용자 비활성화** 를 선택합니다. **Enabled** 열에서 값은 **True** 또는 **False** 로 업데이트됩니다.

2.4. 대시보드를 사용하여 사용자 삭제

다른 사용자를 삭제하려면 관리자 역할이 있는 사용자여야 합니다. 이 작업은 되돌릴 수 없습니다.

절차

1. 대시보드에서 관리자로서 **ID > 사용자를 선택합니다**.
2. 삭제할 사용자를 선택합니다.
3. **사용자 삭제를** 클릭합니다. **사용자 삭제 확인** 창이 표시됩니다.
4. **Delete Users** 를 클릭하여 작업을 확인합니다.

3장. 역할 관리

RHOSP(Red Hat OpenStack Platform)는 역할 기반 액세스 제어(RBAC) 메커니즘을 사용하여 리소스에 대한 액세스를 관리합니다. 역할은 사용자가 수행할 수 있는 작업을 정의합니다. 기본적으로 두 가지 사전 정의된 역할이 있습니다.

- 프로젝트에 연결할 멤버 역할입니다.
- 관리자가 아닌 사용자가 환경을 관리할 수 있도록 하는 관리자 역할.

참고

ID 서비스(keystone)도 역할 목록에 표시될 **reader** 역할을 추가했습니다. Secure RBAC를 활성화한 경우에만 **reader** 역할을 사용합니다.

환경과 관련된 사용자 지정 역할을 생성할 수도 있습니다.

3.1. RED HAT OPENSTACK PLATFORM 관리자 역할 이해

사용자에게 **admin** 역할을 할당하면 이 사용자에게 모든 프로젝트의 리소스를 확인, 변경, 생성 또는 삭제할 수 있는 권한이 있습니다. 이 사용자는 공개적으로 사용 가능한 Glance 이미지 또는 공급자 네트워크와 같이 프로젝트 전체에서 액세스할 수 있는 공유 리소스를 생성할 수 있습니다. 또한 **admin** 역할이 있는 사용자는 사용자를 생성하거나 삭제하고 역할을 관리할 수 있습니다.

admin 역할을 사용자에게 할당하는 프로젝트는 **openstack** 명령이 실행되는 기본 프로젝트입니다. 예를 들어 **development** 라는 프로젝트의 **admin** 사용자가 다음 명령을 실행하면 **development** 프로젝트에 **internal-network** 라는 네트워크가 생성됩니다.

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

admin 사용자는 **--project** 매개변수를 사용하여 모든 프로젝트에서 **internal-network** 를 생성할 수 있습니다.

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

3.2. CLI를 사용하여 역할 보기

관리자는 기존 역할의 세부 정보를 볼 수 있습니다.

절차

1. 사전 정의된 역할을 나열합니다.

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

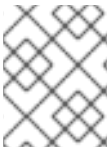
```
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

2. 지정된 역할의 세부 정보를 표시합니다.

```
$ openstack role show admin
```

예제

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



참고

각 역할과 연결된 권한에 대한 자세한 정보를 얻으려면 각 API 호출에 대한 액세스를 감사해야 합니다. 자세한 내용은 API 액세스 감사를 참조하십시오.

3.3. CLI를 사용하여 역할 생성 및 할당

관리자는 다음 명령 세트와 함께 Identity 서비스(keystone) 클라이언트를 사용하여 역할을 생성하고 관리할 수 있습니다. 각 Red Hat OpenStack Platform 배포에는 하나 이상의 프로젝트, 사용자 한 개, 하나의 역할이 연결되어 있어야 합니다.

두 개 이상의 프로젝트에 사용자를 할당할 수 있습니다. 사용자를 여러 프로젝트에 할당하려면 역할을 생성하고 사용자-프로젝트 쌍에 해당 역할을 할당합니다.

참고

이름 또는 ID를 사용하여 사용자, 역할 또는 프로젝트를 지정할 수 있습니다.

절차

1. **new-role** 역할을 생성합니다.

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

2. 프로젝트에 사용자를 할당하려면 먼저 다음 명령을 사용하여 사용자, 역할, 프로젝트 이름 또는 ID를 찾습니다.
   - OpenStack 사용자 목록
   - OpenStack 역할 목록
   - OpenStack 프로젝트 목록
3. 사용자 프로젝트 쌍에 역할을 할당합니다.

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

다음 예제에서는 **demo** 프로젝트의 **admin** 사용자에게 **admin** 역할을 할당합니다.

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

4. 사용자 **admin** 의 역할 할당을 확인합니다.

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

다음 예제에서는 **admin** 사용자가 **admin** 역할이 있는 **demo** 프로젝트에 할당되었는지 확인합니다.

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

3.4. 암시적 역할 생성

ID 서비스(keystone)는 사용자가 특정 역할에 할당되었는지 확인하도록 액세스 제어를 적용합니다. ID 서비스는 암시적 역할 할당을 사용합니다. 사용자를 명시적으로 역할에 할당하는 경우 사용자를 암시적으로 추가 역할에 할당할 수도 있습니다. Red Hat OpenStack Platform에서 기본 암시적 역할을 볼 수 있습니다.

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

참고

ID 서비스(keystone)도 역할 목록에 표시될 **reader** 역할을 추가했습니다. Secure RBAC를 활성화한 경우에만 **reader** 역할을 사용합니다.

권한이 높은 역할은 권한이 적은 역할과 연결된 권한을 의미합니다. 위의 기본 암시적 역할에서 admin은 member를 의미하고, 멤버는 reader를 의미합니다. 암시적 역할을 사용하면 사용자의 역할 할당이 누적 처리되므로 사용자가 하위 역할을 상속합니다.

사용자 지정 역할을 사용하는 경우 암시적 연결을 생성할 수 있습니다.

참고

새 역할을 생성하면 기본적으로 **멤버** 역할과 동일한 액세스 정책이 적용됩니다. 사용자 지정 역할에 대한 고유한 정책 생성에 대한 자세한 내용은 액세스 제어를 위해 정책 파일 사용을 참조하십시오.

절차

- 다음 명령을 사용하여 다른 역할을 나타내는 역할을 지정합니다.

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

검증

- 암시적인 모든 역할을 나열합니다.

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

오류가 발생한 경우 변경 사항을 취소할 수 있습니다.

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

4장. 그룹 관리

ID 서비스(keystone) 그룹을 사용하여 여러 사용자 계정에 일관된 권한을 할당할 수 있습니다.

4.1. CLI를 사용하여 그룹 구성

그룹을 생성하고 그룹에 권한을 할당합니다. 그룹 멤버는 그룹에 할당하는 것과 동일한 권한을 상속합니다.

1. **grp-Auditors**: 그룹을 생성합니다.

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

2. keystone 그룹 목록을 확인합니다.

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

3. **member** 역할을 사용하는 동안 **grp-Auditors** 그룹에 **demo** 프로젝트에 액세스할 수 있는 권한을 부여합니다.

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

4. 기존 사용자 **user1** 을 **grp-Auditors** 그룹에 추가합니다.

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

5. **user1** 이 **grp-Auditors**:의 멤버인지 확인합니다.

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

6. **user1** 에 할당된 유효한 권한을 검토합니다.

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

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

4.2. 대시보드를 사용하여 그룹 구성

대시보드를 사용하여 keystone 그룹의 멤버십을 관리할 수 있습니다. 그러나 명령줄을 사용하여 그룹에 역할 권한을 할당해야 합니다. 자세한 내용은 CLI를 사용하여 그룹 구성을 참조하십시오.

4.2.1. 그룹 생성

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Groups**를 선택합니다.
3. **+Create Group** 을 클릭합니다.
4. 그룹에 대한 이름 및 설명을 입력합니다.
5. **그룹 생성**을 클릭합니다.

4.2.2. 그룹 멤버십 관리

대시보드를 사용하여 keystone 그룹의 멤버십을 관리할 수 있습니다.

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Groups**를 선택합니다.
3. 편집하려는 그룹의 **멤버 관리**를 클릭합니다.
4. **사용자 추가** 를 사용하여 그룹에 사용자를 추가합니다. 사용자를 제거하려면 확인란을 표시하고 **사용자 제거**를 클릭합니다.

5장. 할당량 관리

클라우드 관리자는 프로젝트의 할당량을 설정하고 관리할 수 있습니다. 각 프로젝트에는 리소스가 할당되며 프로젝트 사용자에게 이러한 리소스를 사용할 수 있는 액세스 권한이 부여됩니다. 이를 통해 여러 프로젝트에서 서로의 권한 및 리소스를 방해하지 않고 단일 클라우드를 사용할 수 있습니다. 새 프로젝트를 생성하면 리소스 할당량 집합이 사전 구성됩니다. 할당량에는 프로젝트에 할당할 수 있는 인스턴스, RAM 및 유동 IP의 양이 포함됩니다. 할당량은 프로젝트와 project-user 수준에서 모두 적용할 수 있습니다. 대시보드를 사용하여 새 프로젝트 및 기존 프로젝트에 대해 Compute 및 Block Storage 할당량을 설정하거나 수정할 수 있습니다. 자세한 내용은 프로젝트 관리를 참조하십시오.

5.1. 사용자의 COMPUTE 할당량 보기

다음 명령을 실행하여 사용자의 현재 설정된 할당량 값을 나열합니다.

절차

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

예제

```
$ 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. 사용자의 컴퓨팅 할당량 업데이트

다음 명령을 실행하여 특정 할당량 값을 업데이트합니다.

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

예제

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

참고

quota-update 명령의 옵션 목록을 보려면 다음을 실행합니다.

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

5.3. 사용자의 오브젝트 스토리지 할당량 설정

오브젝트 스토리지 할당량은 다음 카테고리로 분류할 수 있습니다.

- 컨테이너 할당량 - 단일 컨테이너에 저장할 수 있는 총 크기(바이트) 또는 오브젝트 수를 제한합니다.
- 계정 할당량 - 사용자가 Object Storage 서비스에서 사용할 수 있는 총 크기(바이트)를 제한합니다.

컨테이너 할당량 또는 계정 할당량을 설정하려면 Object Storage 프록시 서버에 **proxy-server.conf** 파일의 **[pipeline:main]** 섹션에 추가된 **container_quotas** 또는 **account_quotas** (또는 둘 다) 매개 변수가 있어야 합니다.

```
[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
```

다음 명령을 사용하여 Object Storage 할당량을 보고 업데이트합니다. 프로젝트에 포함된 모든 사용자는 프로젝트에 배치된 할당량을 볼 수 있습니다. 프로젝트에서 오브젝트 스토리지 할당량을 업데이트하려면 프로젝트에서 ResellerAdmin의 역할이 있어야 합니다.

계정 할당량을 보려면 다음을 수행합니다.

```
# 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
```

할당량을 업데이트하려면 다음을 수행합니다.

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

예를 들어 계정에 5GB 할당량을 배치하려면 다음을 수행합니다.

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

6장. 프로젝트 관리

클라우드 관리자는 프로젝트를 생성하고 관리할 수 있습니다. 프로젝트는 OpenStack 사용자 및 그룹을 할당할 수 있는 공유 가상 리소스 풀입니다. 각 프로젝트에서 공유 가상 리소스의 할당량을 구성할 수 있습니다. 서로의 권한 및 리소스를 방해하지 않는 Red Hat OpenStack Platform으로 여러 프로젝트를 생성할 수 있습니다. 사용자는 둘 이상의 프로젝트와 연결할 수 있습니다. 각 사용자에게는 할당된 각 프로젝트에 대해 역할이 할당되어야 합니다.

6.1. 프로젝트 생성

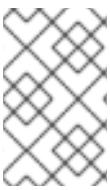
프로젝트를 생성하고 프로젝트에 멤버를 추가하고 프로젝트의 리소스 제한을 설정합니다.

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Projects**를 선택합니다.
3. **프로젝트 만들기**를 클릭합니다.
4. **프로젝트 정보** 탭에서 프로젝트에 대한 이름 및 설명을 입력합니다. **Enabled** 확인란은 기본적으로 선택됩니다.
5. **프로젝트 멤버** 탭의 **모든 사용자** 목록에서 프로젝트에 멤버를 추가합니다.
6. **Quotas** 탭에서 프로젝트에 대한 리소스 제한을 지정합니다.
7. **프로젝트 만들기**를 클릭합니다.

6.2. 프로젝트 편집

프로젝트를 편집하여 이름 또는 설명을 변경하거나, 활성화 또는 일시적으로 비활성화하거나, 프로젝트의 멤버를 업데이트할 수 있습니다.

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Projects**를 선택합니다.
3. 프로젝트 **작업** 열에서 화살표를 클릭하고 **프로젝트 편집을** 클릭합니다.
4. **프로젝트 편집** 창에서 프로젝트를 업데이트하여 이름 또는 설명을 변경하고 프로젝트를 활성화하거나 일시적으로 비활성화할 수 있습니다.
5. **프로젝트 멤버** 탭에서 프로젝트에 멤버를 추가하거나 필요에 따라 제거합니다.
6. **저장**을 클릭합니다.



참고

Enabled 확인란은 기본적으로 선택됩니다. 프로젝트를 일시적으로 비활성화하려면 **Enabled** 확인란을 지웁니다. 비활성화된 프로젝트를 활성화하려면 **Enabled** 확인란을 선택합니다.

6.3. 프로젝트 삭제

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.

2. **Identity > Projects**를 선택합니다.
3. 삭제할 프로젝트를 선택합니다.
4. **Delete Projects**를 클릭합니다. **Confirm Delete Projects** 창이 표시됩니다.
5. **Delete Projects**를 클릭하여 작업을 확인합니다.

프로젝트가 삭제되고 모든 사용자 쌍이 연결 해제됩니다.

6.4. 프로젝트 할당량 업데이트

할당량은 클라우드 리소스를 최적화하기 위해 각 프로젝트에 대해 설정한 운영 제한입니다. 프로젝트 리소스가 알림 없이 사용되지 않도록 할당량을 설정할 수 있습니다. 프로젝트와 프로젝트 수준 모두에서 할당량을 적용할 수 있습니다.

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Projects**를 선택합니다.
3. 프로젝트 **작업** 열에서 화살표를 클릭하고 **할당량 수정**을 클릭합니다.
4. 필요에 따라 **Quota** 탭에서 프로젝트 할당량을 수정합니다.
5. **저장**을 클릭합니다.

참고

현재 *중첩된 할당량*은 아직 지원되지 않습니다. 따라서 프로젝트 및 하위 프로젝트에 대해 할당량을 개별적으로 관리해야 합니다.

6.5. 활성 프로젝트 변경

대시보드를 사용하여 프로젝트의 오브젝트와 상호 작용할 수 있도록 프로젝트를 활성 프로젝트로 설정합니다. 프로젝트를 활성 프로젝트로 설정하려면 프로젝트 멤버여야 합니다. 또한 사용자가 두 개 이상의 프로젝트의 멤버여야 하며 **Set as Active Project** 옵션을 활성화해야 합니다. 비활성화된 프로젝트를 다시 활성화하지 않는 한 활성 상태로 설정할 수 없습니다.

1. 관리 권한이 있는 사용자로 대시보드에 로그인합니다.
2. **Identity > Projects**를 선택합니다.
3. 프로젝트 **작업** 열에서 화살표를 클릭하고 **Set as Active Project**를 클릭합니다.
4. 또는 관리자가 아닌 사용자로 프로젝트 **작업** 열에서 **Set as Active Project**를 클릭하여 열의 기본 작업이 됩니다.

6.6. 프로젝트 계층 구조

Identity 서비스(keystone)에서 멀티 테넌시를 사용하여 프로젝트를 중첩할 수 있습니다. 멀티 테넌시를 사용하면 하위 프로젝트가 상위 프로젝트에서 역할 할당을 상속할 수 있습니다.

6.6.1. 계층적 프로젝트 및 하위 프로젝트 생성

keystone 도메인 및 프로젝트를 사용하여 Hierarchical Multitenancy(HMT)를 구현할 수 있습니다. 먼저 새

도메인을 생성한 다음 해당 도메인에 프로젝트를 생성합니다. 그런 다음 해당 프로젝트에 하위 프로젝트를 추가할 수 있습니다. 해당 하위 프로젝트의 **admin** 역할에 사용자를 추가하여 하위 프로젝트의 관리자로 사용자를 승격할 수도 있습니다.

참고

keystone에서 사용하는 HMT 구조는 현재 대시보드에 표시되지 않습니다.

절차

1. **corp** 라는 새 keystone 도메인을 생성합니다.

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

2. **corp** 도메인에 상위 프로젝트(**private-cloud**)를 생성합니다.

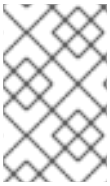
```
$ openstack project create private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d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

3. **corp** 도메인도 지정하는 동안 **private-cloud** 상위 프로젝트 내에 하위 프로젝트(**dev**)를 생성합니다.

```
$ 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** 라는 다른 하위 프로젝트를 생성합니다.

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



참고

Identity API를 사용하여 프로젝트 계층을 볼 수 있습니다. 자세한 내용은 https://developer.openstack.org/api-ref/identity/v3/index.html?expanded=show-project-details-detail에서 참조하십시오.

6.6.2. 계층적 프로젝트에 대한 액세스 구성

기본적으로 새로 생성된 프로젝트에는 역할이 할당되지 않습니다. 상위 프로젝트에 역할 권한을 할당하는 경우 **--inherited** 플래그를 포함하여 하위 프로젝트에서 상위 프로젝트에서 할당된 권한을 상속하도록 지시할 수 있습니다. 예를 들어 상위 프로젝트에 대한 **관리자** 역할 액세스 권한이 있는 사용자는 하위 프로젝트에 대한 **관리자** 액세스 권한도 있습니다.

사용자에게 액세스 권한 부여

1. 프로젝트에 할당된 기존 권한을 확인합니다.

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

2. 기존 역할을 확인합니다.

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

3. 사용자 계정 **user1** 에 **private-cloud** 프로젝트에 대한 액세스 권한을 부여합니다.

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

--inherited 플래그를 사용하여 이 명령을 다시 실행합니다. 결과적으로 **user1** 은 역할 할당을 상속한 **private-cloud** 하위 프로젝트에도 액세스할 수 있습니다.

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

4. 권한 업데이트 결과를 검토합니다.

```
$ 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 사용자는 **qa** 및 **dev** 프로젝트에 대한 액세스 권한을 상속했습니다. 또한 **--inherited** 플래그가 상위 프로젝트에 적용되었으므로 **user1** 은 나중에 생성된 모든 하위 프로젝트에 대한 액세스도 수신합니다.

사용자의 액세스 제거

명시적 및 상속된 권한은 별도로 제거해야 합니다.

1. 명시적으로 할당된 역할에서 사용자를 제거합니다.

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

2. 변경 사항의 결과를 검토합니다. 상속된 권한은 여전히 존재합니다.

```
$ 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. 상속된 권한을 제거합니다.

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

4. 변경 사항의 결과를 검토합니다. 상속된 권한이 제거되었으며 결과 출력이 비어 있습니다.

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

6.6.3. 리셀러 프로젝트 개요

리셀러 프로젝트의 목표는 도메인 계층 구조를 갖는 것입니다. 이러한 도메인을 사용하면 결국 완전한 클라우드를 나타내는 하위 도메인을 사용하여 클라우드의 일부를 재판매하는 것을 고려할 수 있습니다. 이 작업은 다음과 같이 1단계로 나뉘어져 있습니다.

리셀러의 1 단계

리셀러(상위 1)는 여기에 설명된 Hierarchical Multitenancy(HMT)의 확장입니다. 계층적 프로젝트 및 하위 프로젝트 생성. 이전에는 keystone 도메인은 원래 데이터베이스 백엔드에 고유한 테이블을 사용하여 사용자 및 프로젝트를 저장하는 컨테이너로 설계되었습니다. 결과적으로 도메인은 이제 자체 테이블에 더 이상 저장되지 않으며 프로젝트 테이블에 병합되었습니다.

- 도메인은 이제 **is_domain** 플래그로 구분되는 프로젝트 유형입니다.
- 도메인은 프로젝트 계층 구조의 최상위 프로젝트를 나타냅니다. 도메인은 프로젝트 계층 구조에서 루트입니다.
- **projects** 하위 경로를 사용하여 도메인을 생성하고 검색하도록 API가 업데이트되었습니다.
  - **is_domain** 플래그가 true로 설정된 프로젝트를 생성하여 새 도메인을 생성합니다.
  - domain: **is_domain** 쿼리 매개 변수를 포함하여 프로젝트를 가져옵니다.

6.7. 프로젝트 보안 관리

보안 그룹은 프로젝트 인스턴스에 할당할 수 있고 인스턴스에 대한 네트워킹 액세스를 정의하는 IP 필터 규칙 집합입니다. 보안 그룹은 프로젝트에 따라 다릅니다. 프로젝트 멤버는 보안 그룹에 대한 기본 규칙을 편집하고 새 규칙 세트를 추가할 수 있습니다.

모든 프로젝트에는 기타 정의된 보안 그룹이 없는 인스턴스에 적용되는 기본 보안 그룹이 있습니다. 기본값을 변경하지 않는 한 이 보안 그룹은 들어오는 모든 트래픽을 거부하고 인스턴스에서 나가는 트래픽만 허용합니다.

인스턴스를 생성하는 동안 또는 실행 중인 인스턴스의 포트에 보안 그룹을 직접 적용할 수 있습니다.

참고

인스턴스를 생성하는 동안 RBAC(역할 기반 액세스 제어) 공유 보안 그룹을 인스턴스에 직접 적용할 수 없습니다. 인스턴스에 RBAC-공유 보안 그룹을 적용하려면 먼저 포트를 만들고, 공유 보안 그룹을 해당 포트에 적용한 다음 해당 포트를 인스턴스에 할당해야 합니다. 포트에 보안 그룹 추가를 참조하십시오.

필요한 송신을 허용하는 그룹을 생성하지 않고 기본 보안 그룹을 삭제하지 마십시오. 예를 들어 인스턴스에서 DHCP 및 메타데이터를 사용하는 경우 인스턴스에 DHCP 서버 및 메타데이터 에이전트로 송신할 수 있는 보안 그룹 규칙이 필요합니다.

6.7.1. 보안 그룹 생성

보안 규칙을 구성할 수 있도록 보안 그룹을 만듭니다. 예를 들어 ICMP 트래픽을 활성화하거나 HTTP 요청을 비활성화할 수 있습니다.

절차

1. 대시보드에서 **Project > Compute > Access & Security**를 선택합니다.
2. **Security Groups** 탭에서 **Create Security Group** 을 클릭합니다.
3. 그룹에 대한 이름 및 설명을 입력하고 **Create Security Group** 을 클릭합니다.

6.7.2. 보안 그룹 규칙 추가

기본적으로 새 그룹의 규칙은 발신 액세스만 제공합니다. 추가 액세스를 제공하려면 새 규칙을 추가해야 합니다.

절차

1. 대시보드에서 **Project > Compute > Access & Security**를 선택합니다.
2. **보안 그룹** 탭에서 편집하려는 보안 그룹에 대한 **규칙 관리를** 클릭합니다.
3. **Add Rule** 을 클릭하여 새 규칙을 추가합니다.
4. 규칙 값을 지정하고 **추가** 를 클릭합니다.
   다음 규칙 필드는 필수입니다.

   Rule

   규칙 유형. 규칙 템플릿(예: 'SSH')을 지정하면 해당 필드가 자동으로 채워집니다.

   - TCP: 일반적으로 시스템 간에 데이터를 교환하고 최종 사용자 통신에 사용됩니다.
   - UDP: 일반적으로 시스템, 특히 애플리케이션 수준에서 데이터를 교환하는 데 사용됩니다.
   - ICMP: 일반적으로 라우터와 같은 네트워크 장치에서 오류 메시지를 보내거나 메시지를 모니터링하는 데 사용됩니다.

   방향

   Ingress(inbound) 또는 Egress(outbound).

   포트 열기

   TCP 또는 UDP 규칙의 경우 **포트** 또는 **포트 범위** (단일 포트 또는 포트 범위)를 엽니다.

   - 포트 범위의 경우 **From Port** 및 **To Port** 필드에 포트 값을 입력합니다.
   - 단일 포트의 경우 포트 필드에 **포트** 값을 입력합니다.

   유형

   ICMP 규칙 유형은 '-1:255' 범위에 있어야 합니다.

   코드

   ICMP 규칙에 대한 코드는 '-1:255' 범위에 있어야 합니다.

   원격

   이 규칙의 트래픽 소스:

   - CIDR(Classless Inter-Domain Routing): 블록 내의 IP에 대한 액세스를 제한하는 IP 주소 블록입니다. 소스 필드에 CIDR을 입력합니다.

- 보안 그룹: 그룹의 모든 인스턴스가 다른 그룹 인스턴스에 액세스할 수 있도록 하는 소스 그룹입니다.

6.7.3. 보안 그룹 규칙 삭제

더 이상 필요하지 않은 보안 그룹 규칙을 삭제합니다.

절차

1. 대시보드에서 **Project > Compute > Access & Security**를 선택합니다.
2. **보안 그룹** 탭에서 보안 그룹에 **대한 규칙 관리를** 클릭합니다.
3. 보안 그룹 규칙을 선택하고 **Delete Rule** 을 클릭합니다.
4. **Delete Rule** 을 다시 클릭합니다.

참고

삭제 작업을 취소할 수 없습니다.

6.7.4. 보안 그룹 삭제

더 이상 필요하지 않은 보안 그룹을 삭제합니다.

절차

1. 대시보드에서 **Project > Compute > Access & Security**를 선택합니다.
2. **보안 그룹** 탭에서 그룹을 선택하고 **Delete Security Groups** 를 클릭합니다.
3. **Delete Security Groups** 를 클릭합니다.

참고

삭제 작업을 취소할 수 없습니다.

7장. 도메인 관리

ID 서비스(keystone) 도메인은 keystone에서 생성할 수 있는 추가 네임스페이스입니다. keystone 도메인을 사용하여 사용자, 그룹 및 프로젝트를 분할합니다. 다른 LDAP 또는 Active Directory 환경에서 사용자를 인증하도록 이러한 별도의 도메인을 구성할 수도 있습니다. 자세한 내용은 ID 서비스 통합 가이드를 참조하십시오.

참고

ID 서비스에는 **Default** 라는 기본 제공 도메인이 포함되어 있습니다. 이 도메인을 서비스 계정에 대해서만 예약하고 사용자 계정에 대해 별도의 도메인을 생성하는 것이 좋습니다.

7.1. 도메인 목록 보기

openstack domain list 명령을 사용하여 도메인 목록을 볼 수 있습니다.

```
$ 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. 새 도메인 생성

openstack domain create 명령을 사용하여 새 도메인을 생성할 수 있습니다.

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

7.3. 도메인의 세부 정보 보기

openstack domain show 명령을 사용하여 도메인의 세부 정보를 볼 수 있습니다.

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

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

7.4. 도메인 비활성화

요구 사항에 따라 도메인을 비활성화하고 활성화할 수 있습니다.

절차

1. **--disable** 옵션을 사용하여 도메인을 비활성화합니다.

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

2. 도메인이 비활성화되었는지 확인합니다.

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

3. 필요한 경우 **--enable** 옵션을 사용하여 도메인을 다시 활성화합니다.

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

8장. 애플리케이션 인증 정보

애플리케이션 인증 정보를 사용하여 구성 파일에 사용자 계정 자격 증명을 포함하는 것을 방지할 수 있습니다. 대신 사용자는 단일 프로젝트에 대한 위임된 액세스를 수신하고 고유한 시크릿을 보유한 애플리케이션 자격 증명을 생성합니다. 사용자는 위임된 권한을 해당 프로젝트의 단일 역할로 제한할 수도 있습니다. 이를 통해 인증된 서비스가 모든 프로젝트와 역할이 아닌 하나의 프로젝트와 역할에 대해서만 액세스할 수 있는 최소 권한 원칙을 채택할 수 있습니다.

이 방법론을 사용하여 사용자 자격 증명을 공개하지 않고 API를 사용할 수 있으며 애플리케이션은 포함된 사용자 인증 정보 없이도 Keystone에 인증할 수 있습니다.

애플리케이션 인증 정보를 사용하여 토큰을 생성하고 애플리케이션의 **keystone_authtoken** 설정을 구성할 수 있습니다. 이러한 사용 사례는 다음 섹션에 설명되어 있습니다.

참고

애플리케이션 자격 증명은 생성한 사용자 계정에 따라 달라지므로 해당 계정이 삭제된 경우 종료되거나 관련 역할에 대한 액세스 권한이 끊어집니다.

8.1. 애플리케이션 인증 정보를 사용하여 토큰 생성

애플리케이션 인증 정보는 대시보드에서 셀프 서비스 기능으로 사용자가 사용할 수 있습니다. 이 예제에서는 사용자가 애플리케이션 인증 정보를 생성한 다음 이를 사용하여 토큰을 생성하는 방법을 보여줍니다.

1. 테스트 프로젝트를 생성하고 사용자 계정을 테스트합니다.
   a. **AppCreds** 라는 프로젝트를 생성합니다.

   ```
   $ openstack project create AppCreds
   ```

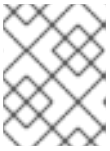
   b. **AppCredsUser** 라는 사용자를 생성합니다.

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

   c. **AppCreds** 프로젝트의 **멤버** 역할에 **AppCredsUser** 액세스 권한을 부여합니다.

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

2. 대시보드에 **AppCredsUser** 로 로그인하고 애플리케이션 인증 정보를 만듭니다.
   개요 → ID → 애플리케이션 인증 정보 → + 애플리케이션 인증 정보 생성.



참고

애플리케이션 자격 증명 이라는 팝업 창을 닫은 후 **clouds.yaml** 파일 내용을 다시 다운로드할 수 없는지 확인합니다.

3. CLI를 사용하여 **/home/stack/.config/openstack/clouds.yaml** 이라는 파일을 생성하고 **clouds.yaml** 파일의 내용을 붙여넣습니다.

```
# 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"
```

참고

이러한 값은 배포에 따라 다릅니다.

4. 애플리케이션 자격 증명을 사용하여 토큰을 생성합니다. 다음 명령을 사용할 때 특정 사용자로 소싱해서는 안 되며 **clouds.yaml** 파일과 동일한 디렉터리에 있어야 합니다.

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

참고

***init*()과 유사한 오류가 발생하면 'application_credential_secret'이라는 키워드 인수가 예기치 않은** 경우 이전 인증 정보로 계속 제공될 수 있습니다. 새로운 환경의 경우 **sudo su - stack** 을 실행합니다.

8.2. 애플리케이션과 애플리케이션 인증 정보 통합

애플리케이션 자격 증명을 사용하여 keystone에 애플리케이션을 인증할 수 있습니다. 애플리케이션 자격 증명을 사용하는 경우 **keystone_authtoken** 설정은 **v3applicationcredential** 을 인증 유형으로 사용하고 인증 정보 생성 프로세스 중에 받은 인증 정보를 포함합니다. 다음 값을 입력합니다.

- **application_credential_secret**: 애플리케이션 인증 정보 시크릿입니다.
- **application_credential_id**: 애플리케이션 인증 정보 ID입니다.
- (선택 사항) **application_credential_name**: ID가 아닌 이름이 지정된 애플리케이션 인증 정보를 사용하는 경우 이 매개변수를 사용할 수 있습니다.

예를 들면 다음과 같습니다.

```
[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. 애플리케이션 인증 정보 관리

명령줄을 사용하여 애플리케이션 인증 정보를 생성하고 삭제할 수 있습니다.

create 하위 명령은 현재 소싱된 계정을 기반으로 애플리케이션 자격 증명을 생성합니다. 예를 들어 **admin** 사용자로 가져올 때 인증 정보를 생성하면 애플리케이션 인증 정보에 동일한 역할이 부여됩니다.

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

주의

--unrestricted 매개변수를 사용하면 애플리케이션 인증 정보가 다른 애플리케이션 인증 정보 및 신뢰를 생성하고 삭제할 수 있습니다. 이는 잠재적으로 위험한 동작이며 기본적으로 비활성화되어 있습니다. **--unrestricted** 매개변수는 다른 액세스 규칙과 함께 사용할 수 없습니다.

기본적으로 결과 역할 멤버십에는 자격 증명을 생성한 계정에 할당된 모든 역할이 포함됩니다. 특정 역할에만 액세스 권한을 위임하여 역할 멤버십을 제한할 수 있습니다.

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

애플리케이션 인증 정보를 삭제하려면 다음을 수행합니다.

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

8.4. 애플리케이션 인증 정보 교체

애플리케이션 인증 정보는 사용자 계정을 생성한 사용자 계정에 바인딩되고 사용자 계정이 삭제되거나 사용자가 위임된 역할에 대한 액세스 권한이 손실된 경우 유효하지 않게 됩니다. 따라서 필요에 따라 새 애플리케이션 인증 정보를 생성할 준비가 되어 있어야 합니다.

구성 파일의 기존 애플리케이션 인증 정보 교체

애플리케이션에 할당된 애플리케이션 인증 정보를 업데이트합니다(구성 파일을 사용하여).

1. 새 애플리케이션 자격 증명 세트를 생성합니다.
2. 기존 자격 증명을 교체하여 애플리케이션 구성 파일에 새 자격 증명을 추가합니다. 자세한 내용은 애플리케이션과 애플리케이션 인증 정보 통합을 참조하십시오.
3. 애플리케이션 서비스를 다시 시작하여 변경 사항을 적용합니다.
4. 필요한 경우 이전 애플리케이션 인증 정보를 삭제합니다. 명령줄 옵션에 대한 자세한 내용은 애플리케이션 인증 정보 관리를 참조하십시오.

clouds.yaml에서 기존 애플리케이션 인증 정보 교체

clouds.yaml 에서 사용하는 애플리케이션 인증 정보를 교체할 때 OpenStack 사용자 인증 정보를 사용하여 교체 자격 증명을 생성해야 합니다. 기본적으로 애플리케이션 자격 증명을 사용하여 다른 애플리케이션 인증 정보 세트를 생성할 수 없습니다. **openstack application credential create** 명령은 현재 소싱된 계정을 기반으로 애플리케이션 인증 정보를 생성합니다.

1. 만료될 인증 자격 증명을 처음 생성한 OpenStack 사용자로 인증합니다. 예를 들어 애플리케이션 인증 정보를 사용하여 토큰을 생성하는 절차를 사용한 경우 **AppCredsUser** 로 다시 로그인해야 합니다.
2. **AppCred2** 라는 애플리케이션 자격 증명을 만듭니다. 이 작업은 OpenStack 대시보드 또는 **openstack** CLI 인터페이스를 사용하여 수행할 수 있습니다.

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

3. 이전 명령의 출력에서 **id** 및 **secret** 매개변수를 복사합니다. **secret** 매개변수 값은 다시 액세스할 수 없습니다.

4. **${HOME}/.config/openstack/clouds.yaml** 파일의 **application_credential_id** 및 **application_credential_ secret 매개변수 값을 복사한 시크릿** 및 **id** 값으로 바꿉니다.

검증

1. clouds.yaml을 사용하여 토큰을 생성하여 인증 정보가 예상대로 작동하는지 확인합니다. 다음 명령을 사용할 때 특정 사용자로 소싱해서는 안 되며 **clouds.yaml** 파일과 동일한 디렉터리에 있어야 합니다.

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

출력 예:

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