Enabling Cloud-Native Applications with Application Credentials in Keystone

Colleen Murphy
Cloud Developer at SUSE
Overview

- Why we needed application credentials
- What are application credentials? (with demo!)
- The future of application credentials
Before...
Cloud applications

```python
from cinderclient import client
from keystoneauth1 import session
from keystoneauth1.identity.generic import password

auth = password.Password(username='cmurphy',
                          password='secrets',
                          project_name='production',
                          user_domain_name='LDAP_EMEA',
                          project_domain_name='Default',
                          auth_url='https://cloud.example.com/identity')

s = session.Session(auth=auth)
cinder = client.Client('3', session=s)
cinder.volume_backups.create('5ee22c66-4ce7-4136-bffa-371a4cf40d43')
```
Principle of Least Privilege

- Applications have access to everything the user has access to
Passwords in config files

- openrc files
- clouds.yaml
- {nova,cinder,neutron,...}.conf
- yourapplication.ini

Protecting plaintext secrets: https://review.openstack.org/474304
LDAP passwords in config files

LDAP domain

admin role on

creates

app user

application domain

Bug 968696

cmurray user
Password rotation == downtime

Steps to change a keystone user's compromised password:
1. openstack user set --password moresecurepassword appuser
2. [applications are suddenly down, being unable to authenticate]
3. Update config files on all worker nodes
4. Restart services on all worker nodes
5. [applications can auth again]
Introducing Application Credentials
Application Credentials

An application credential is a **scoped** auth method that a user creates to delegate a subset of their role assignments on a single project to something else - whoever or whatever possesses knowledge of the identifier and the secret belonging to the application credential.

- Has its own secret
- Can only access one project, no matter how many projects the user is in
- Can have all or a subset of the roles the user has on that project
- Is **user-lived** - when the user is deleted, the app credential dies
- User can have many
What's in a name?

Why are they called application credentials? What's wrong with API keys?

- "Application credentials" is a name we invented without any industry-known connotations
Why not trusts?

- Not fully self-service
- Still requires your keystone user's password to auth
Live demo
Authenticating

clouds:
opstack:
  auth:
    auth_url: https://cloud.example.com/identity/v3
    application_credential_id: "a2911c0aadea457e8d713955ab3675d0"
    application_credential_secret: "BB6L1wghFcr5AlZ3JK6vE1-B936vACEJJoof"
    region_name: "RegionOne"
    interface: "public"
    identity_api_version: 3
    auth_type: "v3applicationcredential"
Authenticating

clouds:
openstack:
  auth:
    auth_url: https://cloud.example.com/identity/v3
    username: "cmurphy"
    user_domain_name: "suse.de"
    application_credential_name: "volume_backups_001"
    application_credential_secret: "BB6L1wghFcr5AlZ3JK6vE1-B936vACEJJ0of"
    region_name: "RegionOne"
    interface: "public"
    identity_api_version: 3
    auth_type: "v3applicationcredential"
Rotation

1. openstack application credential create volume_backups_cred_002
2. [applications are still using old app cred]
3. Update config files on all worker nodes
4. Restart services on all worker nodes [applications start using the new app cred]
5. openstack application credential delete volume_backups_cred_001
What about project-lived credentials?

The need:

- Team member writes an application for a keystone project
- Creates application credential for the project, shared with the team
- Team member is reassigned
- Application keeps working
What about project-lived credentials?

The problem:

- Employee privately creates application credential for a keystone project, records secret
- Employee's keystone user is deleted
- Employee can still access that project using the application credential identifier and secret
Handling team attrition

If the team member that created the application credential is leaving:
Plan ahead. Rotate the application credential before their user is decommissioned in order to avoid downtime.

If someone else on the team is leaving:
Plan ahead! For security, the application credential should still be rotated, even though the user leaving won't cause downtime.

Keystone can't solve people problems.
The Future
Fine-grained access control

Currently:
openstack application credential create myappcred \
   --role member

Soon:
openstack application credential create myappcred \
   --capabilities \
   '[["service": "volume", "path": "/v3/{project_id}/backups", "type": "POST"]]'
Rotation automation

Automating around user-lived application credentials
System scope

Allow cloud administrators to automate system-level tasks
Thanks! Questions?

#openstack-keystone
openstack-dev mailing list

cmurphy
@_colleenm