caso Documentation

Release

Spanish National Research Council (CSIC)

October 18, 2016

Contents

cASO is a pluggable extractor of Cloud Accounting Usage Records from an OpenStack installation. cASO gets usage information from nova or ceilometer APIs and can generate valid output for Apel SSM or logstash.

Contents:

CHAPTER 1

Installation

1.1 Pre-requisites

If you are planning to use cASO for generating accounting records for EGI, you will need a valid APEL/SSM configuration. Follow the documentation available at the EGI FedCloud wiki

1.2 Installation

At the command line:

```
$ pip install caso
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv caso
$ pip install caso
```

1.2.1 CentOS 6

On CentOS 6, you can use Software Collections to install Python 2.7:

```
$ yum -y install centos-release-SCL
$ yum -y install python27
```

There are also some dependencies of the packages used by cASO that need to be installed (gcc, libffi-devel adn openssl-devel):

```
$ yum -y install gcc libffi-devel openssl-devel
```

You can then install pip for that version of Python and use that to install cASO:

```
$ scl enable python27 bash
$ easy_install-2.7 pip
$ pip install caso
$ exit    # this terminates bash with the SCL python2.7
```

In this case you can later on use caso-extract with the following command line:

```
$ scl enable python27 caso-extract
```

Alternatively, if you want to use a virtualenv:

```
$ scl enable python27 bash
$ virtualenv caso
$ . caso/bin/activate
$ pip install caso
$ exit    # this terminates bash with the SCL python2.7
```

Running from the virtualenv:

```
$ scl enable python27 caso/bin/caso-extract
```

CHAPTER 2

Configuration

2.1 cASO configuration

cASO uses a config file (default at /etc/caso/caso.conf) with several sections. A sample file is available at etc/caso/caso.conf.sample.

2.1.1 [DEFAULT] section

The [DEFAULT] section configures the basic behavior of cASO. The sample config file (/etc/caso/caso.conf.sample) includes a description of every option. You should check at least the following options:

- extractor (default value: nova), specifies which extractor to use for getting the data. The following APIs are supported: ceilometer and nova. Both should generate equivalent information.
- site_name (default value: <None>). Name of the site as defined in GOCDB.
- tenants (list value, default empty). List of the tenants to extract records from.
- messengers (list, default: caso.messenger.noop.NoopMessenger). List of the messengers to publish data to. APEL messenger is: caso.messenger.ssm.SsmMessager, LogStash is caso.messenger.logstash.LogstashMessenger

2.1.2 [extractor] section

This section specifies the configuration of the extractor (mainly the credentials to connect to the API). Check the following:

- user (default: accounting), name of the user. This user needs proper permission to query the API for the tenant usages.
- password (default: None), password of the user.
- endpoint (default: None), keystone endpoint to authenticate with.
- mapping_file (default: /etc/caso/voms.json). File containing the mapping from VOs to local tenants as configured in Keystone-VOMS, in the form:

```
    {
        "VO": {
            "tenants": ["foo", "bar"],
```

```
        }
    }
```

- `insecure` (default: `False`), wether to check or not the server's certificate.

Important: Your should not use `insecure=True` in production! If you get a SSL error (`CERTIFICATE_VERIFY_FAILED`), this is probably due to the fact that the request module CA bundle does not contain the CA of your server.

If you are using the request module of your distribution package, it is normally patched to use the system's default CA bundle (`/etc/ssl/certs/ca-certificates.crt` from the `ca-certificates` package on Debian systems and `/etc/pki/tls/certs/ca-bundle.crt` from the ‘‘ca-certificates‘‘on RH systems). Check the packages documentation to add a new CA to those bundles.

If you are not installing request through the distribution packages (e.g. via pip), it uses its own vendorized CA bundle, located in the distribution directory (i.e. requests/cacert.pem). It should be enough to append the correct certificates to the end of the cacert.pem file. In a virtualenv, the bundle should be located at `$VIRTUAL_ENV/lib/python2.7/site-packages/requests/`

2.1.3 `[ssm]` section

Options defined here configure the SSM messenger. There is only one option at the moment:

- `output_path` (default: `/var/spool/apel/outgoing/openstack`), directory to put the generated SSM records. APEL/SSM should be configured to take records from that directory.

2.1.4 `[logstash]` section

Options defined here configure the logstash messenger. Available options:

- `host` (default: `localhost`), host of Logstash server.
- `port` (default: `5000`), Logstash server port.

2.2 OpenStack Configuration

The user configured in the previous section has to be a member of each of the tenants (another option is to convert that user in an administrator, but the former option is a safer approach) for which it is extracting the accounting. Otherwise, `cASO` will not be able to get the usages and will fail:

```
keystone role-create --name accounting
keystone user-create --name accounting --pass <password>
# For each of the tenants, add the user with the accounting role
keystone user-role-add --user accounting --role accounting --tenant <tenant>
```

Also, this user needs access to Keystone so as to extract the users information.

- If you are using the V2 identity API, you have to give admin rights to the `accounting` user, editing the `/etc/keystone/policy.json` file and replacing the line:

```
    "admin_required": "role:admin or is_admin:1 or",
```

  with:

```
    "admin_required": "role:admin or is_admin:1 or role:accounting",
```

- If you are using the V3 identity API you can grant the user just the rights for listing the users adding the appropriate rules in the `/etc/keystone/policy.json`.

CHAPTER 3

Usage

3.1 command line

cASO provides the `caso-extract` command to generate new records from your OpenStack deployment. `caso-extract -h` will show a complete list of available arguments.

Use the `--extract_from` argument to specify the date from when the records should be extracted. If no value is set, then cASO will extract the records from the last run. If equal to "None", then extract records from the beggining of time. If not time zone is specified, UTC will be used.

Important: If you are running an OpenStack Nova version lower than Kilo there is a bug in its API, making impossible to paginate over deleted results.

Since nova is limiting the results to 1000 by default, if you are expecting more than 1000 results you will get just the last 1000. This is important if you are publishing data for the first time, or if you are republishing all your accounting). If this is your case, adjust the *osapi_max_limit* to a larger value in */etc/nova/nova.conf*.

3.2 Running as a cron job

The best way of running cASO is via a cron job like the following:

```
10 * * * * caso-extract
```

3.3 Migration from OSSSM

If you had a previous installation of osssm, you can migrate to cASO following these steps:

1. Remove the previous osssm installation (e.g. remove apel-ssm-openstack rpm).
2. Remove any cron jobs related to `osssm.extract` or `osssm.push`, a single cron job as described above is enough. You should keep the cron job that executes `ssmsend`, this is still needed to send the records to the accounting database.