CloudKitty

CloudKitty

CloudKitty is the project name of a Rating-as-a-Service project.

Motivation

From the beginning of the Ceilometer project billing and pricing were excluded of the goals to achieve. But since those aspects are inherent to providing cloud (for both public and/or private), this gap needed to be filled using both the same technologies and the same Open Source approach than the rest of every OpenStack project. That is the goal of CloudKitty which is so far developed by Objectif Libre \[1\] and Catalyst IT \[2\], but we are really hoping to get more developers on board on a next future.

The project started before the Atlanta's Summit, and a prototype was shown to few people (mainly from the Ceilometer project), they all gave us great feedback and were interested in the project.

We know that billing is made of generic rules, but also some specific ones, most billing operations are business specific. Our approach is to be modular allowing a huge flexibility in the treatments. We will focus on many generic rules, but allowing specific cases to be ruled by specific modules.

CloudKitty has been made to be highly modular on 4 levels:

- input data sources (collectors)
- rating policies (rating pipeline)
- output storage (storage)
- output file format (writers, used to generate reports)

CloudKitty supports multiple collectors, multiple rating policies and multiple outputs.

An overview of CloudKitty has been demoed during the OpenStack Summit in Paris, here is the video https://www.openstack.org/summit/openstack-paris-summit-2014/session-videos/presentation/sponsor-demo-theater-objective-libre-cloudkitty-open-source-rating-component-for-openstack

How to use it?

CloudKitty proposes the following way to interact with it:

- Horizon
- API
- Python Binding
- Python client (in progress)
How to install CloudKitty
Follow the installation documentation on developers documentation \[3\].
If you want to quickly test CloudKitty, you can use devstack \[4\].

Architecture
CloudKitty has been designed to use the same modules and architecture of other OpenStack components, such as:
- oslo.db
- oslo.config
- pecan
- WSME
- stevedore

Licencing
CloudKitty is released under Apache 2.0 Licence.

Current Implementation
The project is under high work in progress, but the basis of the architecture are already present including the dynamic modular architecture that allows to enable/disable modules (and thus capabilities / treatment) on the fly.

Current Collectors
A huge focus has been put on integrating the OpenStack metrics from Ceilometer, hence the standard Ceilometer collector module. So far only the compute, image, volume and network collectors has been implemented.

Current REST API
The v1 API is documented on developers documentation \[5\].

Current Horizon Integration
The horizon integration falls in 2 parts:
- administration
- user

Administration View
Using the administration pane, it is possible to activate / deactivate services. It is also possible to pass values to configure the various services.
To get an idea of the horizon integration for the administrators we have published a video that shows the management of the price for resources by an admin: https://www.youtube.com/watch?v=KlagCqTUPco
User View

It is displaying "real-time" price (the granularity is defined in the administration panel).

To get an idea of the horizon integration from an user perspective we have published a video that shows the user part of CloudKitty:

• How the user will see the estimated price of the resource they are about to launch https://www.youtube.com/watch?v=CmaBXzv28oI
• How the user see the total estimed price he will be charged for his past usage https://www.youtube.com/watch?v=v6m1vPl55pg

RoadMap

Some aspects are moving very fast, here is the content of our RoadMap.

General Architecture

Since we want to spread the project, it is important to enable translation in the various parts of CloudKitty.

RoadMap of the REST API

The roadmap here is focusing on the API of each new module, and storage backend query API.

RoadMap of the Horizon Integration

The roadmap here is focusing on configuring each new module using Horizon.

Documentation

We have an automatically generated documentation (extracted from the code) available at http://docs.openstack.org/developer/cloudkitty/

Community

Sources

• Cloudkitty [6]
• Python Cloudkitty client library [7]
• Horizon plugin for CloudKitty [8]

IRC

#cloudkitty #cloudkitty [9] on freenode

References

[9] irc://irc.freenode.net/
Article Sources and Contributors


License

Attribution 3.0 Unported (CC BY 3.0)
http://creativecommons.org/licenses/by/3.0/