Transitioning your Cloud Monitoring from Ceilometer to Monasca

Witek Bedyk
Joseph Davis

Who we are

Witek Bedyk, SUSE

IRC: witek

Joseph Davis, SUSE

IRC: joadavis

Agenda

- Motivation
- History
- Architecture
- Transition Use Cases
- Why should I transition?
- Try it for yourself

Motivation

- State of the Telemetry project
  - State of Gnocchi project - unmaintained
- Generic monitoring solution
  - Infrastructure metrics
  - Application metrics
  - Monitoring-as-a-Service
  - Logging
  - Events (in development)
- Encourage cooperation between two projects

History

Ceilometer History

Ceilometer is an old service in OpenStack. Ceilometer dates back to before the Folsom OpenStack release.

Officially moved from StackForge to OpenStack in 2012.

Ceilometer started as “a unique point of contact for billing systems to acquire all of the measurements they need to establish customer billing, across all current OpenStack core components”.

Through contributions and development, Ceilometer added support for more services as well as Event Handling and Alarming.

Ceilometer History

As Ceilometer grew to include more metrics, and as time passed and samples piled up, it became apparent that the original database backends did not keep up.

In 2014, the Gnocchi project was started to create a Time-Series database for metric storage. It was designed to be scalable, multi-tenant, and platform agnostic.

Around that time, Aodh was split out for alarming, and Panko for Event handling in the Liberty release

All these sub-projects were created under the Telemetry project.

Ceilometer History

Gnocchi was moved out of OpenStack in June 2017.

As of the Queens release, the Telemetry project had lost most developers and development on Aodh and Panko stopped.

Further reading - https://julien.danjou.info/lessons-from-openstack-telemetry-deflation/

As of now, Gnocchi is unmaintained - https://github.com/gnocchixyz/gnocchi/issues/1049

Fortunately, in the Train cycle a new leadership has stepped up for Telemetry and support continues. (Thank you Trinh Nguyen, Rong Zhu, and others)

Monasca History

2013
Monasca project started
by HP and Rackspace

11.2014
First presentation at OpenStack Summit
Monasca in production

11.2015
Logs support added
Monasca accepted as the official OpenStack project

1.2017
Monasca containerized

11.2018
Monasca in Kolla

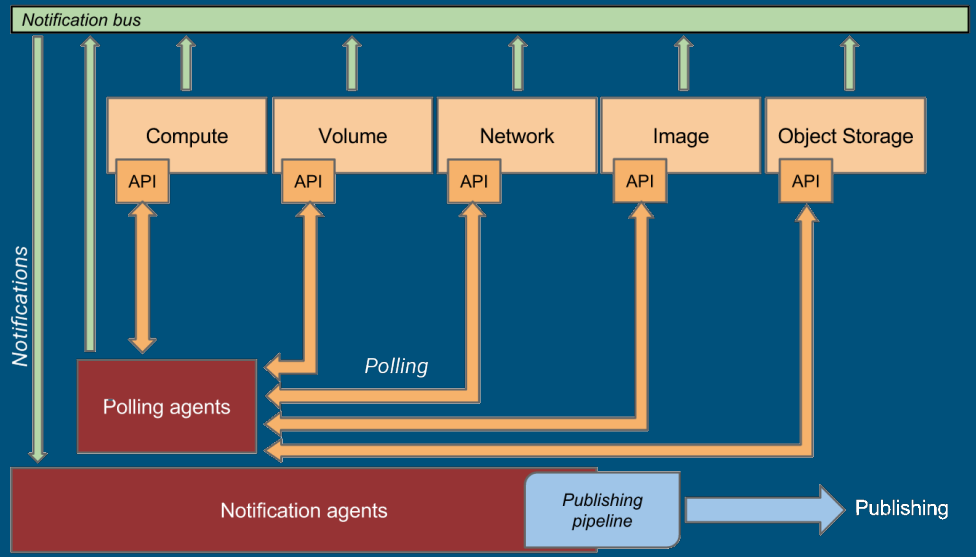
Architecture

Gathering Metrics (Ceilometer)

Samples collected by notification or polling agents

Plug in architecture for adding new measurement sources

Only OpenStack samples



Gathering Metrics (Monasca)

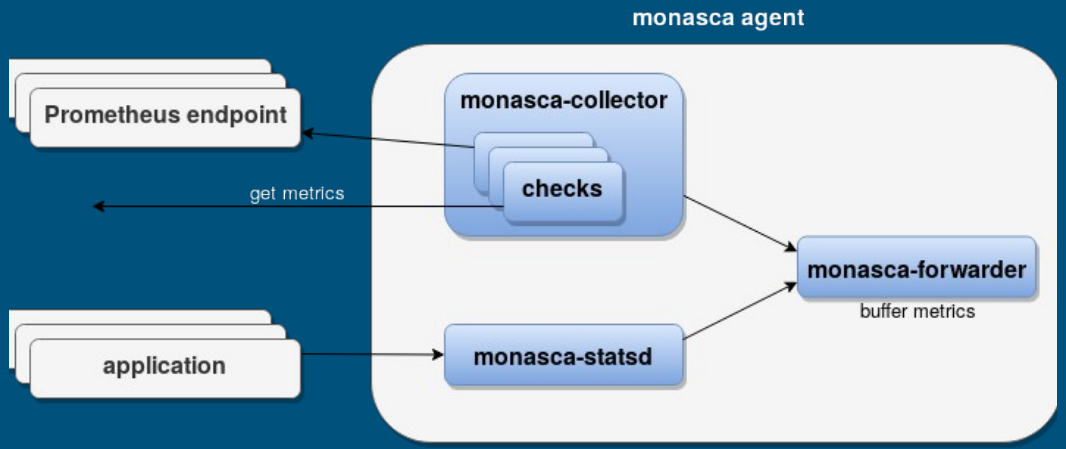
Collector periodically gathers **system** and **application** metrics.

Pluggable, e.g. *Libvirt, OVS, Ceph*

Can scrape *Prometheus* endpoints

StatsD daemon

Forwarder buffers measurements



Gathering Metrics (Monasca)

Cloud users and administrators can collect application metrics:

- Instrument application with StatsD client (push model)
- Instrument application with Prometheus client (pull model)
- Use/implement own Monasca Agent plugin
- Use/implement Prometheus exporter
- Post measurements directly to Monasca API

Publishing Metrics (Ceilometer)

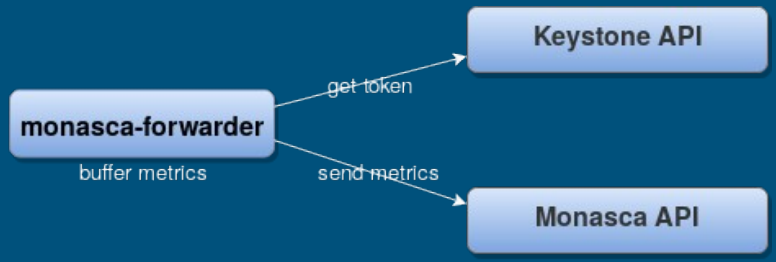
Samples sent to central Ceilometer storage service (Gnocchi is the only *officially* supported configuration for measurements up through Train)

Publishing Metrics (Monasca)

Forwarder builds batches of measurements

Measurements buffered in case API unavailable

Measurements posted to Monasca API



Storing Metrics (Ceilometer)

Default and the only officially supported backend is Gnocchi.

Gnocchi API can be used to retrieve measurements.

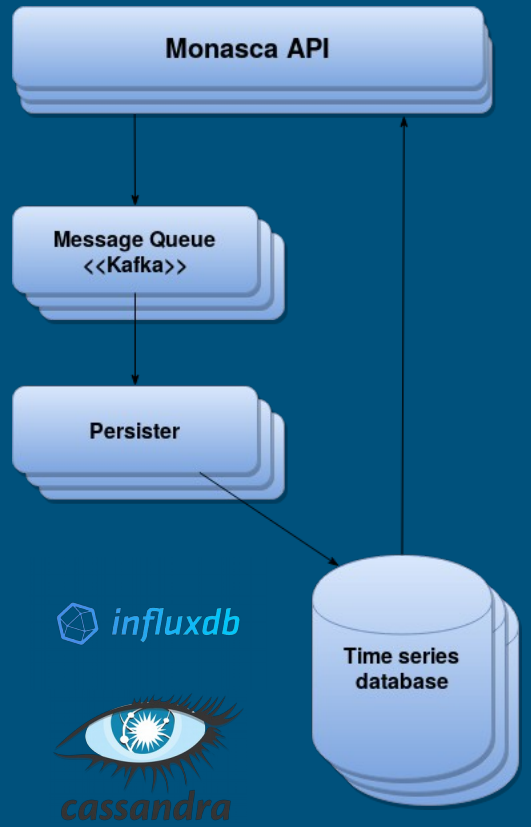
Storing Metrics (Monasca)

Apache Kafka message queue for performance, scalability and resilience

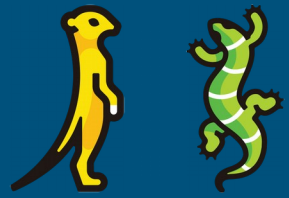
Persister writes to time series database

InfluxDB and Apache Cassandra officially supported backends

Monasca API queries database for measurements and statistics



Ceilosca (Ceilometer and Monasca)



Ceilosca started at HP in 2015.

Some metrics were being generated in Ceilometer but had no equivalent in Monasca, so Ceilosca created a publisher from the Ceilometer Agent to the Monasca API.

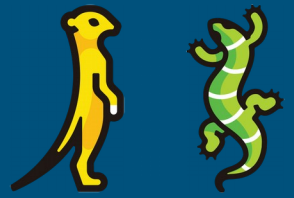
Used in production in multiple product releases.

https://opendev.org/openstack/monasca-ceilometer/

NEW Monasca Publisher in Ceilometer repo in Ussuri release
https://review.opendev.org/#/c/562400/

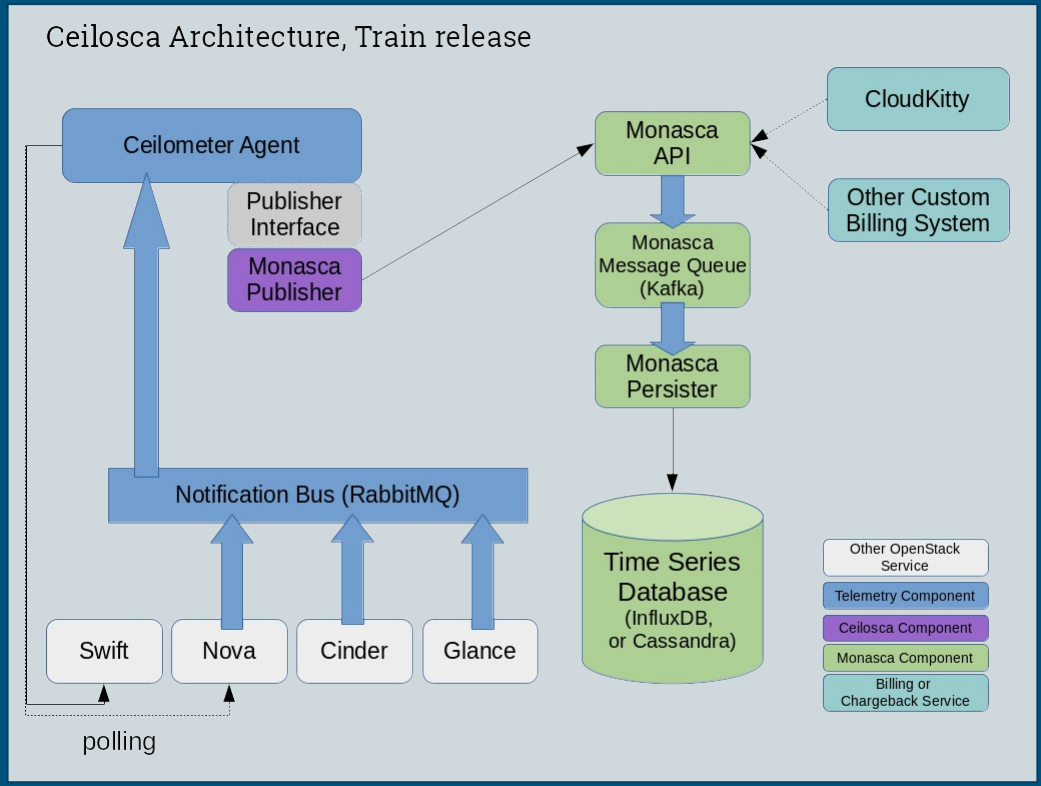
Many thanks to the Telemetry team for reviewing and accepting the Publisher

Ceilosca (Ceilometer and Monasca)



Ceilosca uses the standard Publisher Interface

Can be configured to send any metric Ceilometer collects using standard pipeline configuration options



Architectural Advantages of Monasca

- Buffering
  - Measurements don’t get lost if system is loaded
  - More efficient network usage for measurement reporting
- Scalability
- Using commodity storage options
  - No dependency on Gnocchi

Transition Use Cases

Be Aware

We don’t transfer existing data from Ceilometer to Monasca.

There isn’t a way to export from Gnocchi or another store to Monasca, and the data is in a different, incompatible format.

Before making the change, do a final set of reports or data dump from Ceilometer or Gnocchi as you need for your records.

Transition Use Cases

- New deployment
- Alerting (Alarms and Notifications)
- Rating and Billing
- Auto scaling
- Self Healing
- Visualisation

New Deployment

Simple use case - just deploy Monasca instead of Ceilometer

Monasca integration already part of Kolla

Supports both: deployment as part of Kolla or standalone

https://docs.openstack.org/kolla-ansible/latest/reference/logging-and-monitoring/monasca-guide.html

Alarms and Notifications (Aodh)

Tri-state model: *OK, alarm, insufficient data*

- Threshold based alarms:
  - Static threshold
  - Aggregation function
  - Evaluation period
  - Metadata filtering
- Event based alarms

Queries measurements from Gnocchi

Notification actions: *webhook, log*

Alarms (Monasca)

Thresholding engine is a stream processing application evaluating measurements consumed from the message queue in real-time (sliding windows).

Simple syntax to define alarm definitions (templates). Supports metadata **filtering** and **grouping**.

```
avg(cpu.utilization_perc{scale_group=GROUP_ID}) > 50 times 3
```

Event based alarms in development.

Notifications (Monasca)

Notification engine executes actions defined for alarms triggered by thresholding engine.

Pluggable: *email, webhook, PagerDuty, Slack, Jira*

Rating and Billing

Integration with CloudKitty - OpenStack Rating-as-a-Service

```
# Get the total cost grouped by resource type
$ cloudkitty summary get -g res_type
+------------+---------------+---------+---------------------+---------------------+
| Tenant ID  | Resource Type | Rate    | Begin Time          | End Time            |
+------------+---------------+---------+---------------------+---------------------+
| 9a7[...]87 | image         | 0.30368 | 2018-02-01T00:00:00 | 2018-03-01T00:00:00 |
| 9a7[...]87 | volume        | 0.051   | 2018-02-01T00:00:00 | 2018-03-01T00:00:00 |
+------------+---------------+---------+---------------------+---------------------+
```

https://www.stackhpc.com/cloudkitty-and-monasca-1.html
https://www.objectif-libre.com/en/blog/2018/03/14/integration-monasca-et-cloudkitty/ - and Ceilosca!

Rating and Billing

As with Ceilometer or CloudKitty, Monasca does not provide a feature that reports out dollar amounts and produces invoices to your customers.

But it is possible to use a billing solution that works with CloudKitty.

And it is possible to have a billing solution pull data directly from the Monasca API or using the python-monascaclient as a CLI or programmatic interface.

Note that Prometheus is not suitable for billing (not multi-tenant, does not guarantee delivery)

Auto-Scaling

Auto-Scaling using Heat and Monasca has been possible since Liberty release.

Heat resources for Monasca alarm definition and notification available.

Auto-scaling template example available in **`openstack/heat-templates`** repository.

See the workshop presented at the Denver 2019 Summit
https://github.com/sjamgade/monasca-autoscaling

Integration with other Auto-scaling services in OpenStack are possible via webhook notification (Senlin).

Self Healing

Monasca integrates with multiple OpenStack projects to manage OpenStack infrastructure in a policy-driven fashion, reacting to failures and other events by automatically healing services.

- Congress – policy-based governance
- Vitrage – root cause analysis
- Heat – orchestration
- Watcher – optimization

Visualization - Pretty Graphs

Monasca UI – Horizon plugin

Grafana integration

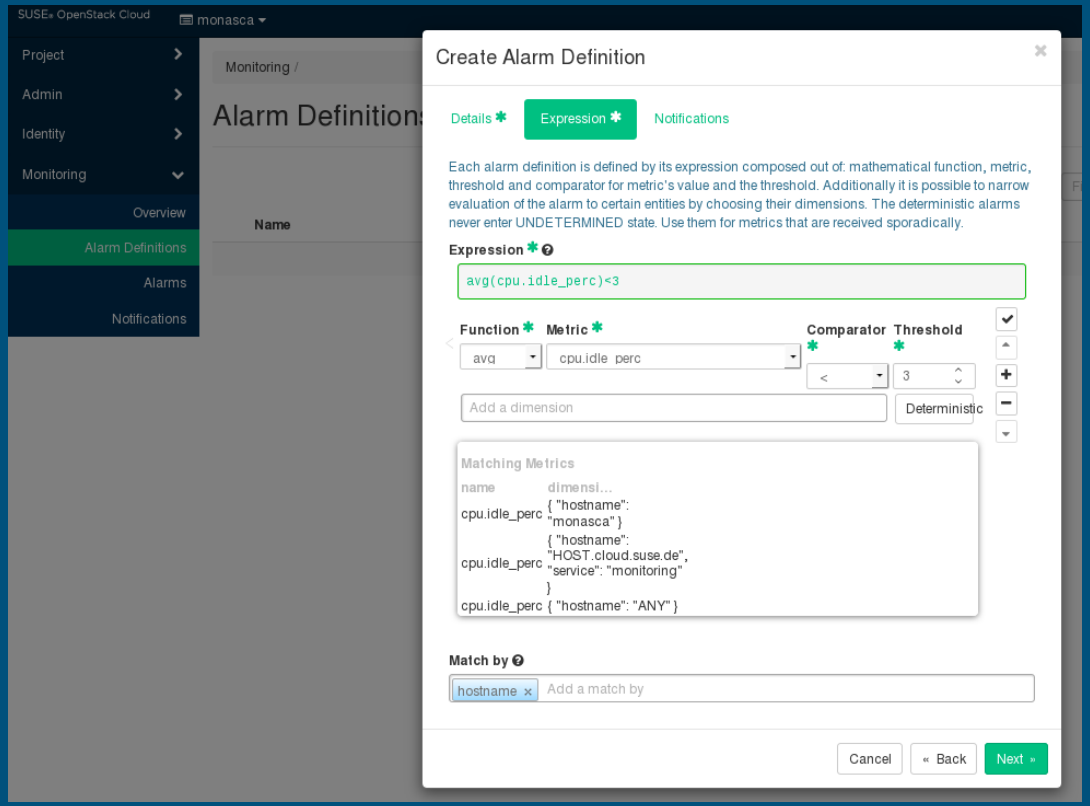
Or export data through Monasca API and have your own visualizer. It is your data, after all.

Horizon plugin

Overview of alarm states

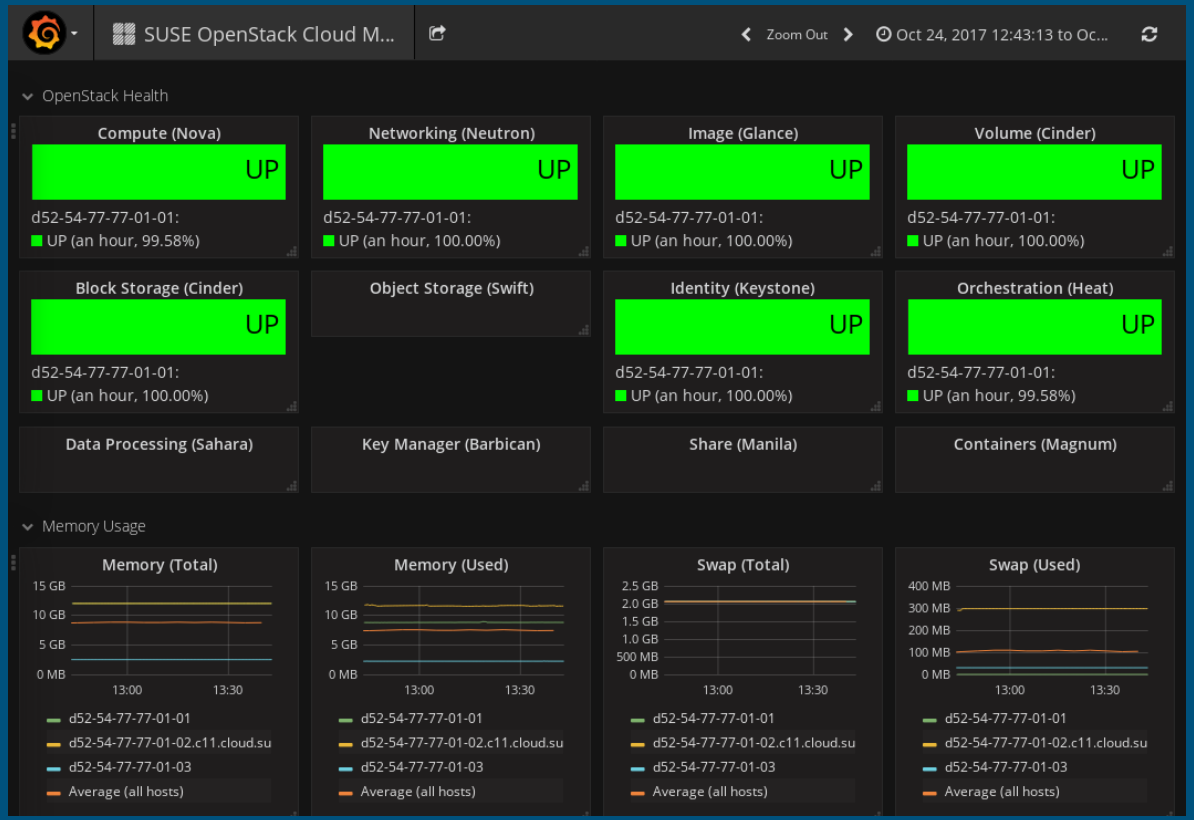
Intuitive interface for defining alarm definition expressions and notification methods.

Integration with Grafana and Kibana



Grafana

Monasca datasource allows creating own dashboards



Logging

Monasca API allows log messages to be gathered to a central service.

Solution based on *Elasticsearch, Logstash and Kibana* (ELK).

Plugins available for *Logstash, FluentD* and *Beaver* for logs collection.

Kibana plugin provides multi-tenancy.

Integration with Horizon dashboard.

Reasons to Transition

	Monasca	Telemetry
multi-tenant	+	+
HA / scalability	+	+
application monitoring	+	-
delivery guarantee	+	-
high-performant	+	-
OpenStack notifications	in development	+
logging	+	-
community	+	-

Call to Action

Try it out for yourself

- DevStack
- Monasca Docker
- Kolla

Get involved in the community

- Email: Use [monasca] in the subject to openstack-discuss@openstack.org
- IRC: at #openstack-monasca

Monasca in Shanghai

- **Monasca - Project Update**
  Tue 9:50am – 5th Floor, 515
- **Efficient Monitoring and Root Cause Analysis in Complex Systems**
  Tue 3:20pm – 6th Floor, 619
- **PTG – Monasca Onboarding**
  Wed 1:30pm – Mitaka

Thank You
谢谢

Questions and Answers

Backup Slides

Architectural comparisons

Ceilometer is for OpenStack samples only, not application data.

Monasca is for Openstack measurements and for generic measurements, such as can be sent from an application.

Monasca offers additional features:

- Centralized collection of logs through Monasca Log API
- Aggregation of metrics using Monasca Transform
- Prometheus integration
- Flexible measurement collection from applications through Monasca API

Additional Feature: Monasca Transform

Monasca Transform is an aggregation engine for Monasca measurements.

Uses the power of Apache Spark to aggregate metrics based on administrator defined rules.

Can combine or summarize metrics and post as new metrics.

More information - https://github.com/openstack/monasca-transform

Ceilometer/Telemetry History

2012
Ceilometer officially moved from StackForge to OpenStack

2013
Contributions added Event Handling and Alarming

2014
Gnocchi project introduced to address storage issues

2015
Aodh and Panko split out in the Liberty release

6.2017
Gnocchi leaves OpenStack

2018
By the Queens release, most developers have left Telemetry Development on Aodh and Panko has stopped

10.2019
Gnocchi is unmaintained

CloudKitty and Ceilosca in one DevStack

```
[[local|localrc]]
# Disable Gnocchi (not needed with Ceilosca)
CEILOMETER_BACKEND=none

# all of Monasca
enable_plugin monasca-api https://opendev.org/openstack/monasca-api.git master

# ceilometer
enable_plugin ceilometer https://opendev.org/openstack/ceilometer.git master

# ceilosca
enable_plugin ceilosca https://opendev.org/openstack/monasca-ceilometer.git master
Enable_service ceilosca

# cloudkitty
enable_plugin cloudkitty https://opendev.org/openstack/cloudkitty.git master
enable_service ck-api,ck-proc
```

CloudKitty and Ceilosca in one DevStack

- *cloudkitty* command line needs Domain information (reference /etc/cloudkitty/cloudkitty.conf for values)
- DevStack deployment does not configure /etc/cloudkitty/metrics.yml to work with Monasca. Have to configure that after deployment.
- DevStack deployment does not assign the ‘monasca-user’ role to cloudkitty user
- DevStack deployment leaves it to user to enable and configure ratings (no default ratings “on” out of the box)

https://www.objectif-libre.com/en/blog/2018/03/14/integration-monasca-et-cloudkitty/
 is a very helpful blog post. Note that some commands have changed syntax since that blog was posted.

Backup Slides - Abstract Submitted for this Session

There has recently been an increase in interest in Monasca as a monitoring service for OpenStack Clouds. Some interest is from new users looking for a scalable service to monitor their cloud. But what if you are already familiar with Ceilometer and the Telemetry project?

We will discuss some of the great things about Monasca and do some comparison of how it can handle some important use cases, such as

- Gathering monitoring data in a cloud
- Sending notifications and alarms
- Supporting pretty graphs
- Support for auto-scaling and self healing
- Providing data for billing (because someone has to pay for your donuts)

What can I expect to learn?

- What features are available from Monasca

- Some comparison of Monasca and Ceilometer architecture and features

- How Monasca can support your monitoring use cases, and even do more than Ceilometer

- What options are available to enable deploying Monasca

- How to get connect with the Monasca community