

# OpenStack get vm cpu\_util with Stein version

Asked 1 year, 9 months ago   Active 1 year, 3 months ago   Viewed 1k times



In stein version, ceilometer remove polling for cpu\_util.

1

Follow this doc: <https://docs.openstack.org/ceilometer/stein/admin/telemetry-measurements.html#openstack-compute>



2

The measurements only cpu ( CPU time used) and vcpus (Number of virtual CPUs allocated to the instance).



And check github commit about ceilometer

<https://github.com/openstack/ceilometer/blob/4ae919c96e4116ab83e5d83f2b726ed44d165278/releases/notes/save-rate-in-gnocchi-66244262bc4b7842.yaml>,

The cpu\_util meters are deprecated.

and this commit about ceilometer remove transformer support.

<https://github.com/openstack/ceilometer/commit/9db5c6c9bfc66018aeb78c4a262e1bfa9b326798#diff-4161ff0e1519a6226d1117b428fc831a>

According to commit message, gnocchi handle transformer data.

So, how to use gnocchi aggregate cpu and vcpus to calculate cpu usage?

virtual-machine

openstack

cpu-usage

metrics

ceilometer

Share   Improve this question   Follow

asked May 20 '19 at 8:08



dommgifer

51 ● 1 ● 1 ● 7

## 1 Answer

Active

Oldest

Votes



Although this question is some three months old, I hope my answer still helps somebody.

2



It seems that Ceilometer's pipeline processing has never worked correctly. As the original poster noticed, the Ceilometer development team took the somewhat drastic measure to deprecate then obsolete this feature. Consequentially, the only CPU meter that remains in the [Ceilometer arsenal](#) is the accumulated CPU time of an instance, expressed in nanoseconds.

To calculate the CPU utilization of a single instance based on this meter, you can use Gnocchi's rate aggregation. If *rate:mean* is one of the aggregation methods in your archive policy, you can do this:

```
gnocchi measures show --resource-id <uuid> --aggregation rate:mean cpu
```

Or use the dynamic aggregation feature for the same result:

```
gnocchi aggregates '(metric cpu rate:mean)' id=<uuid>
```

The first parameter of the *aggregates* command is the **operation** that defines what figures you want. You find operations explained in the [Gnocchi API documentation](#), especially the section that lists [supported operations](#), and the examples section. The second parameter is a **search expression** that limits the calculation to instances with this particular UUID (of course, there is only one such instance).

So far, the command just pulls figures from the Gnocchi database. You can, however, pull the data from the database, *then aggregate them on the fly*. This technique is (or used to be) called *re-aggregation*. This way, you don't need to include *rate:mean* in the archive policy:

```
gnocchi aggregates '(aggregate rate:mean (metric cpu mean))' id=<uuid>
```

The numbers are expressed in nanoseconds, which is a bit unwieldy. Good news: Gnocchi aggregate operations also support arithmetic. To convert nanoseconds to seconds, divide them by one billion:

```
gnocchi aggregates '(/ (aggregate rate:mean (metric cpu mean)) 1000000000)' id=<uuid>
```

And to convert them to percentages, divide them by (granularity times one billion), then multiply the result with 100. Assuming a granularity of 60:

```
gnocchi aggregates '(* (/ (aggregate rate:mean (metric cpu mean)) 60000000000) 100)' id=<uuid>
```

Share Improve this answer Follow

answered Aug 7 '19 at 1:55



[berndbausch](#)

421 ● 3 ● 10