

# Gnocchi



Julien Danjou – *Paris Monitoring Meetup #6* – 18 May 2016

# Hello!



## I am Julien Danjou

Principal Software Engineer at Red Hat

You can find me at @juldanjou



The background is white and decorated with various colorful circles and dashed lines. In the top left, there is a large orange circle with a dashed red outline, overlapping a yellow circle. Below the yellow circle is a small pink circle. In the top center, a large blue number '1' is centered within a large dashed light blue circle. In the top right, there is a green circle with a white dot in the center, a small orange circle, and a yellow circle with a dashed green outline. In the bottom left, there is a green circle with a dashed green outline, a large yellow circle, and a small cyan circle. In the bottom right, there is a large cyan circle with a white dot in the center, and a cyan circle with a dashed blue outline.

# 1

What's the problem?  
And how we solve it.



Storing timeseries and  
resources index

In any infrastructure, you  
have to know what's  
running, for how long, doing  
what. You meter those  
things.

**And then you need to store  
that.**



## Perfect solution

### Scalable

Targeting cloud platforms where thousands of instances and resources pop up every day.


Storing and retrieving data should be fast.

### Easy to use

Provide an API that makes it easy to program against the solution. Build any kind of solution easily (billing, capacity planning, statistical analysis...)

### Easy to operate

Installation and operation should be easy for administrators used to standard UNIX tools.





## Existing solutions

- ◎ Graphite
  - Not scalable
  - Poor code base
  - Not modulable
- ◎ InfluxDB
  - Does not work
  - Does not scale
- ◎ OpenTSDB
  - Need to set up Hadoop

...



## Gnocchi – started in May 2014

### **Part of OpenStack Telemetry**

Designed to solve Ceilometer storage issue back then.

But work stand-alone!

### **Free Software**

Apache Licensed.

### **Easy to install**

`pip install gnocchi`

### **Documented**


Everything is documented. No doc, no merge policy.

### **Written in Python**

With some standard used libraries (SQLAlchemy, Pandas...)

### **Distributed & resilient**

Design to run on cloud platforms. Native high-availability and workload distribution support.



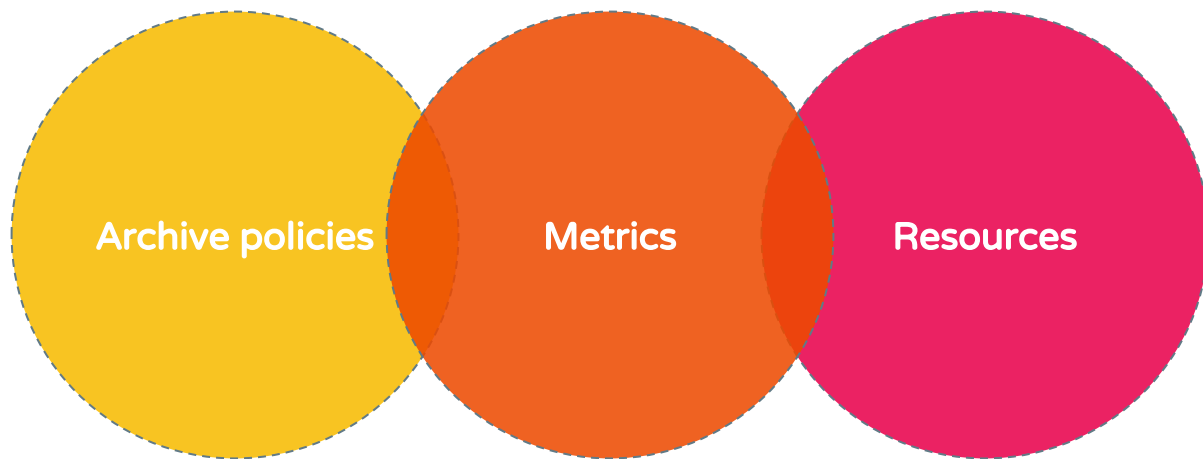
A decorative graphic featuring several overlapping circles in orange, yellow, green, and blue. Some circles have dashed outlines in matching colors. A large, light blue dashed circle is centered in the upper half of the image.

2

How it works  
Basic things you need to know



## Main object types



A decorative background featuring various colored circles (green, yellow, orange, pink, blue) and a large dashed light-blue circle that frames the central text. A teal ring is positioned at the top center.

“

## Archive policy

A measure storage policy attached to a metric. It determines how long measures will be kept in a metric and how they will be aggregated.

*Keep data for:*

- *1 day with 5 minutes granularity*
- *1 month with 1 hour granularity.*

*Compute average, minimum and maximum.*

A decorative background featuring a large, light blue dashed circle. Various colored circles and arcs are scattered around it: a large yellow circle at the top left, a medium blue circle with a white double quote inside at the top center, a large orange circle at the top right, a medium orange circle at the bottom left, a small pink circle at the bottom left, a large yellow circle at the bottom right, a small green circle at the bottom right, and a small blue circle at the bottom right. There are also several small solid circles in green, orange, and pink.

“

## Measures

A datapoint tuple composed of  
timestamp and a value.

**Timestamp:** 2016-05-17T13:43:  
23+0200  
**Value:** 42

A decorative background featuring a large, light blue dashed circle. Various colored circles and arcs are scattered around it: a large yellow circle at the top left, a large cyan arc at the top center, a large orange circle at the bottom left, a large yellow circle at the bottom right, and several smaller circles in green, blue, orange, and pink. A blue circle containing white quotation marks is positioned above the title.

“

## Metric

An entity storing measures identified by an UUID. It can be attached to a resource using a name. How a metric stores its measure is defined by the archive policy it is associated to.

A decorative background featuring a large, light blue dashed circle. Various colored circles and arcs are scattered around it: a large yellow circle at the top left, a small green circle with a white dot at the top left, a small blue circle at the top left, a large orange circle at the bottom left, a small pink circle at the bottom left, a large yellow arc at the bottom left, a large cyan arc at the top right, a small pink circle at the top right, a large yellow circle at the top right, a large orange arc at the top right, a small green circle with a white dot at the bottom right, and a small blue circle at the bottom right.

“

## Resource

An entity representing anything in your infrastructure that you will associate metric with. It is identified by a unique ID and can carry attributes.



3

Demo  
As if you just installed it.



## List archive policies and create a metric

→ gnocchi archive-policy list

name	back_window	definition	aggregation_methods
high	0	- points: 3600, granularity: 0:00:01, timespan: 1:00:00 - points: 10080, granularity: 0:01:00, timespan: 7 days, 0:00:00 - points: 8760, granularity: 1:00:00, timespan: 365 days, 0:00:00	std, count, 95pct, min, max, sum, median, mean
medium	0	- points: 1440, granularity: 0:01:00, timespan: 1 day, 0:00:00 - points: 168, granularity: 1:00:00, timespan: 7 days, 0:00:00 - points: 365, granularity: 1 day, 0:00:00, timespan: 365 days, 0:00:00	std, count, 95pct, min, max, sum, median, mean
low	0	- points: 12, granularity: 0:05:00, timespan: 1:00:00 - points: 24, granularity: 1:00:00, timespan: 1 day, 0:00:00 - points: 30, granularity: 1 day, 0:00:00, timespan: 30 days, 0:00:00	std, count, 95pct, min, max, sum, median, mean

→ gnocchi metric create --archive-policy-name low

Field	Value
archive_policy/aggregation_methods	std, count, 95pct, min, max, sum, median, mean
archive_policy/back_window	0
archive_policy/definition	- points: 12, granularity: 0:05:00, timespan: 1:00:00 - points: 24, granularity: 1:00:00, timespan: 1 day, 0:00:00 - points: 30, granularity: 1 day, 0:00:00, timespan: 30 days, 0:00:00
archive_policy/name	low
created_by_project_id	admin
created_by_user_id	admin
id	95fdc8ff-1aed-4dd3-b65b-bfb53f91081b
name	None
resource/id	None

# Send & retrieve measures

```
→ gnocchi measures add -m 2016-05-16T12:00:00@42 -m 2016-05-16T12:01:03@45 -m 2016-05-16T12:06:07@22 95fdc8ff-1aed-4dd3-b65b-bfb53f91081b
→ gnocchi measures show 95fdc8ff-1aed-4dd3-b65b-bfb53f91081b
```

timestamp	granularity	value
2016-05-16T00:00:00+00:00	86400.0	36.3333333333
2016-05-16T12:00:00+00:00	3600.0	36.3333333333
2016-05-16T12:00:00+00:00	300.0	43.5
2016-05-16T12:05:00+00:00	300.0	22.0

```
→ gnocchi measures show --aggregation min 95fdc8ff-1aed-4dd3-b65b-bfb53f91081b
```

timestamp	granularity	value
2016-05-16T00:00:00+00:00	86400.0	22.0
2016-05-16T12:00:00+00:00	3600.0	22.0
2016-05-16T12:00:00+00:00	300.0	42.0
2016-05-16T12:05:00+00:00	300.0	22.0

```
→ gnocchi measures show --aggregation 95pct 95fdc8ff-1aed-4dd3-b65b-bfb53f91081b
```

timestamp	granularity	value
2016-05-16T00:00:00+00:00	86400.0	44.7
2016-05-16T12:00:00+00:00	3600.0	44.7
2016-05-16T12:00:00+00:00	300.0	44.85
2016-05-16T12:05:00+00:00	300.0	22.0



# Create a resource

```
→ gnocchi resource-type create --attribute name:string --attribute host:string server
```

Field	Value
attributes/host	max_length=255, min_length=0, required=True, type=string
attributes/name	max_length=255, min_length=0, required=True, type=string
name	server

```
→ gnocchi resource create --attribute name:www-42 --attribute host:computel --create-metric cpu:medium --create-metric memory:low --type server `uuidgen`
```

Field	Value
created_by_project_id	admin
created_by_user_id	admin
ended_at	None
host	computel
id	e4c2eab7-52ed-4447-bbcb-48cb04f12015
metrics	cpu: d51d8ba3-ab06-4f0c-af6c-d88dbac8c2a8 memory: 0240ceb8-d1d6-435d-a37c-f7f3bf99a388
name	www-42
original_resource_id	E4C2EAB7-52ED-4447-BBCB-48CB04F12015
project_id	None
revision_end	None
revision_start	2016-05-16T13:35:43.985927+00:00
started_at	2016-05-16T13:35:43.985815+00:00
type	server
user_id	None

# Update a resource

```
→ gnocchi resource update --attribute host:compute2 --type server e4c2eab7-52ed-4447-bbcb-48cb04f12015
```

Field	Value
created_by_project_id	admin
created_by_user_id	admin
ended_at	None
host	compute2
id	e4c2eab7-52ed-4447-bbcb-48cb04f12015
metrics	cpu: d51d8ba3-ab06-4f0c-af6c-d88dbac8c2a8 memory: 0240ceb8-d1d6-435d-a37c-f7f3bf99a388
name	www-42
original_resource_id	E4C2EAB7-52ED-4447-BBCB-48CB04F12015
project_id	None
revision_end	None
revision_start	2016-05-16T13:37:38.140460+00:00
started_at	2016-05-16T13:35:43.985815+00:00
type	server
user_id	None

## See previous updates in JSON

```
→ gnocchi resource history --format json --details e4c2eab7-52ed-4447-bbcb-48cb04f12015
[
  {
    "created_by_user_id": "admin",
    "started_at": "2016-05-16T13:35:43.985815+00:00",
    "user_id": null,
    "revision_end": "2016-05-16T13:37:38.140460+00:00",
    "ended_at": null,
    "created_by_project_id": "admin",
    "metrics": "cpu: d51d8ba3-ab06-4f0c-af6c-d88dbac8c2a8\nmemory: 0240ceb8-d1d6-435d-a37c-f7f3bf99a388",
    "host": "compute1",
    "revision_start": "2016-05-16T13:35:43.985927+00:00",
    "project_id": null,
    "type": "server",
    "id": "e4c2eab7-52ed-4447-bbcb-48cb04f12015",
    "name": "www-42"
  },
  {
    "created_by_user_id": "admin",
    "started_at": "2016-05-16T13:35:43.985815+00:00",
    "user_id": null,
    "revision_end": null,
    "ended_at": null,
    "created_by_project_id": "admin",
    "metrics": "cpu: d51d8ba3-ab06-4f0c-af6c-d88dbac8c2a8\nmemory: 0240ceb8-d1d6-435d-a37c-f7f3bf99a388",
    "host": "compute2",
    "revision_start": "2016-05-16T13:37:38.140460+00:00",
    "project_id": null,
    "type": "server",
    "id": "e4c2eab7-52ed-4447-bbcb-48cb04f12015",
    "name": "www-42"
  }
]
```

# Send & get measures on a metric attached to a resource & search

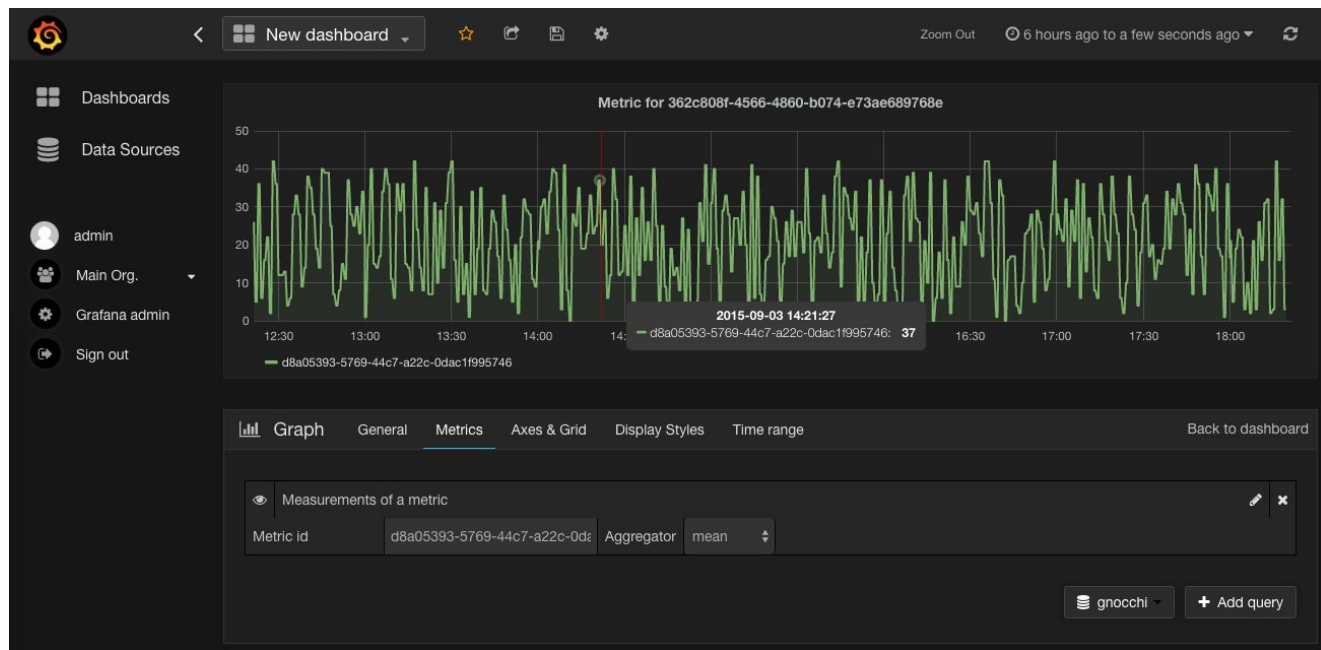
```
→ gnocchi measures add -m 2016-05-16T12:00:00@42 -m 2016-05-16T12:01:03@45 -m 2016-05-16T12:06:07@22 --resource-id e4c2eab7-52ed-4447-bbcb-48cb04f12015 cpu
→ gnocchi measures show --resource-id e4c2eab7-52ed-4447-bbcb-48cb04f12015 cpu
```

timestamp	granularity	value
2016-05-16T00:00:00+00:00	86400.0	36.3333333333
2016-05-16T12:00:00+00:00	3600.0	36.3333333333
2016-05-16T12:00:00+00:00	60.0	42.0
2016-05-16T12:01:00+00:00	60.0	45.0
2016-05-16T12:06:00+00:00	60.0	22.0

```
→ gnocchi resource search --type server host=compute2
```

id	type	project_id	user_id	started_at	ended_at	revision_start	revision_end
e4c2eab7-52ed-4447-bbcb-48cb04f12015	server	None	None	2016-05-16T13:35:43.985815+00:00	None	2016-05-16T13:37:38.140460+00:00	None

# Grafana support





## More awesome features

### Search by metric value, compute aggregations

Look into metrics value and search for outliers.

Compute aggregation across several metrics.

### Batching

Send batch of measures in one single HTTP call.

### Trigger alarms

Using Aodh to evaluate your alarms.

### Compression

Using LZ4 compression to compress data on the fly. Fast, reduce storage usage between x2-5.

### Statsd support


If you're already a Graphite user or you're polling tool support statsd (e.g. collectd), it's compatible.

### Multi-tenant

ACL that guarantees your different tenants can't see each other resources. But the admin can see everything. Customizable.

### HTTP REST API

That's what's used by the 'gnocchi' CLI. Add --debug to discover the HTTP requests, or read the API specs!

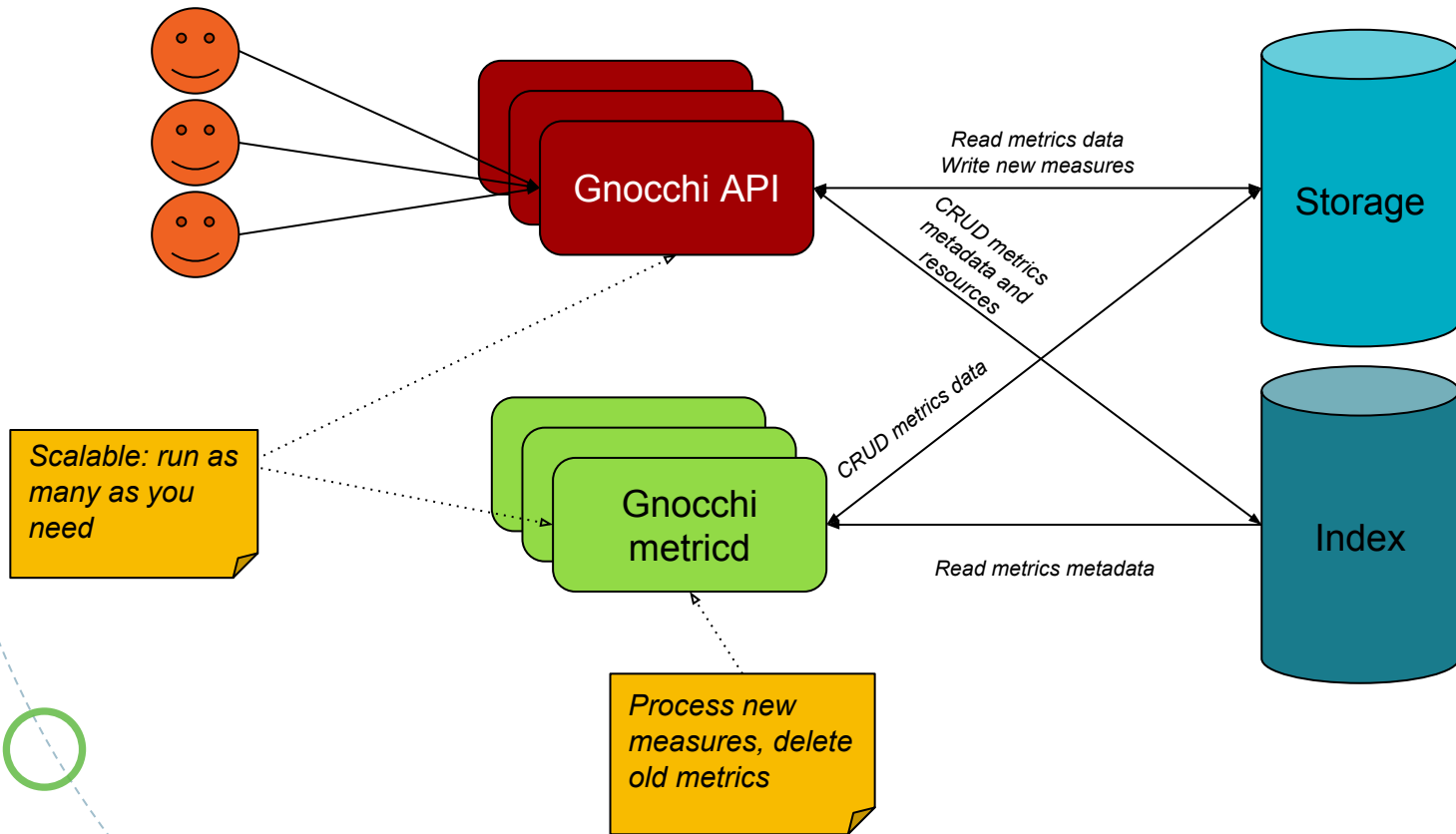


A decorative graphic featuring several overlapping circles in orange, yellow, green, and blue. Some circles have dashed outlines in matching colors. A large, light blue dashed circle is centered in the upper half of the page.

4

Under the hood  
How the magic is working.

# Architecture







## Backends

### Index

Any RDBMS support by SQLAlchemy. Best choice: **PostgreSQL**. Though **MySQL** is also supported.



### Storage

Simple deployment?  
**Plain files** (with NFS if you want).

Scalable and robust?  
Go for **Ceph**.

Got OpenStack?  
Leverage **Swift**.

# Thanks!



<http://gnocchi.xyz>

## Any questions?

You can find me at @juldanjou & [julien@danjou.info](mailto:julien@danjou.info)