

GRAFANACON  
AMS2018

Kub

Dan Cech

[@DanCech](https://twitter.com/DanCech)

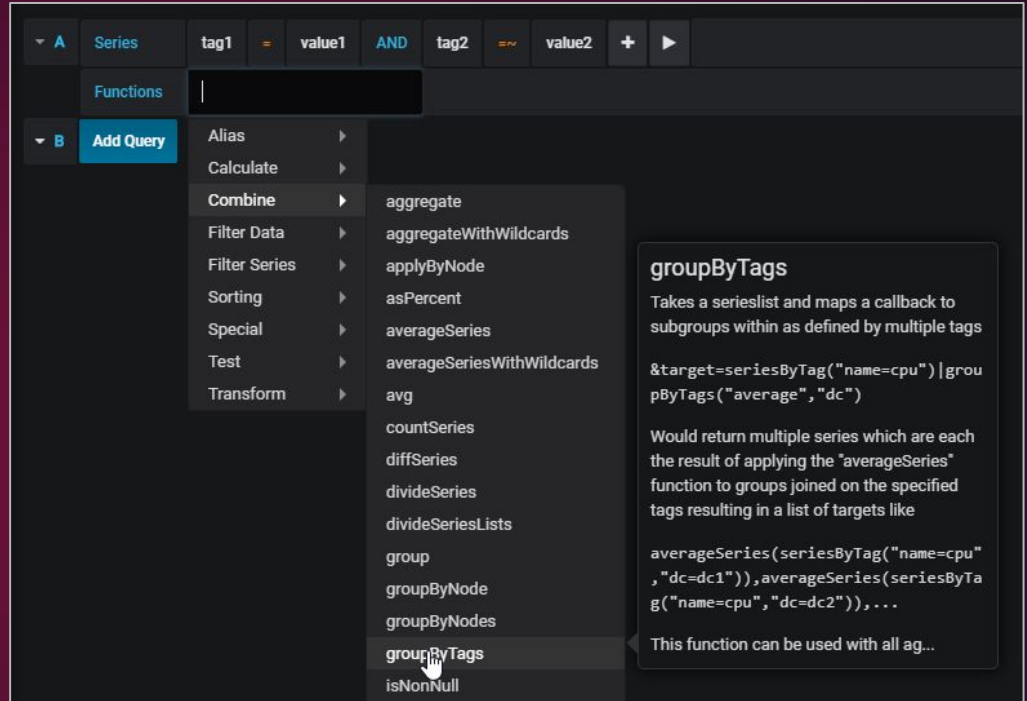
# Teaching an Old Dog New Tricks

- Graphite has been around ~forever (since April 2008)
- I joined the project in 2016
- 2017 was a busy year :)
- 1.0 released (finally!) in April 2017
- 1.1 released in December 2017



# So, What's New?

- Tag Support
- Clustering Updates
- New Function Syntax
- More & Better Functions
- Custom Functions



The screenshot shows the Grafana query editor interface. At the top, a query is defined as `tag1 = value1 AND tag2 = value2`. Below this, the 'Functions' menu is open, showing a list of functions under the 'Add Query' section. The 'groupByTags' function is highlighted. A tooltip for 'groupByTags' is displayed on the right, providing a description and an example query.

**groupByTags**  
Takes a serieslist and maps a callback to subgroups within as defined by multiple tags

`&target=seriesByTag("name=cpu") | groupByTags("average", "dc")`

Would return multiple series which are each the result of applying the "averageSeries" function to groups joined on the specified tags resulting in a list of targets like

`averageSeries(seriesByTag("name=cpu", "dc=dc1")), averageSeries(seriesByTag("name=cpu", "dc=dc2")), ...`

This function can be used with all ag...

# Tag Support

- Old model was purely hierarchical  
`production.dc1.server3.memory.used`
- Inflexible, opaque, brittle
- Tags solve these problems
- What worked in InfluxDB, Prometheus, etc?
- How to build on & maintain compatibility with existing codebase?
- K.I.S.S.

# Tag Support

- Add tags to existing hierarchical scheme

```
memory.used;dc=dc1;env=production;host=server3
```

- Flexible: Add new tags to the naming scheme easily
- Clear: Each element in the scheme has a built-in name
- Resilient: Adding tags doesn't break existing dashboards
- Simple: Uses as much existing tooling as possible
- Pluggable: SQLite, MySQL, Postgres, Redis, HTTP

# Using Tags

- Old: `production.dc1.server3.memory.used`
- New: `seriesByTag(  
 `name=memory.used`,  
 `env=production`,  
 `dc=dc1`,  
 `server=server3`)`
- Supports equality & regex matching `=`, `!=`, `=~` & `!=~`
- Use tag names in alias & group functions

# Clustering

- 1.0 was a major step forward from 0.9
- Rewritten (again) for 1.1
- Streamlined and simplified
- Reduced intra-cluster request volume & wait time
- Added support for msgpack format
- Flexible options for sharding with tags

# New Function Syntax

Old syntax based on nested function calls:

```
alias(movingAverage(sumSeries(api.req.*.cnt), 60), 'api.avg')
```

New syntax defines a pipeline, matches Grafana UI:

```
api.req.*.cnt|sumSeries()|movingAverage(60)|alias('api.avg')
```

Mix and match old and new:

```
movingAverage(api.req.*.cnt|sumSeries(), 60)|alias('api.avg')
```



# More & Better Functions

- Before: many single purpose functions
- Now: smaller number of flexible, consistent functions
- Standard set of aggregation operators:  
average, median, sum, min, max, diff, stddev, count, range, multiply & last
- **Example:** `aggregate` supports all functionality of `averageSeries`, `sumSeries`, `minSeries`, `maxSeries`, `diffSeries`, `stddevSeries`, `countSeries`, `rangeOfSeries` & `multiplySeries` (plus support for `median` & `last`)
- Old function names available as aliases

# Custom Functions

- Extend the Graphite API
- Easy to write!
- Easy to install!
- Instantly available in Grafana
- ASAP Smoothing:

[https://github.com/deniszh/graphite\\_asap](https://github.com/deniszh/graphite_asap)

```
from graphite.functions.params import Param, ParamTypes

def toUpperCase(requestContext, seriesList):
    """Custom function that changes series names to UPPERCASE"""
    for series in seriesList:
        series.name = series.name.upper()
    return seriesList

toUpperCase.group = 'Custom'
toUpperCase.params = [
    Param('seriesList', ParamTypes.seriesList, required=True),
]

SeriesFunctions = {
    'upper': toUpperCase,
}
```

# So, What's Next?

- Prometheus read/write support
- Internal stats reporting
- Tag support in collectd
- Tag support in statsd



GRAFANACÓN  
AMS2018

Thanks!