Science & Technology
Facilities Council

Monitoring with InfluxDB & Grafana

Andrew Lahiff

HEP SYSMAN, Manchester

15th January 2016

Overview

- Introduction
- InfluxDB
- InfluxDB at RAL
- Example dashboards & usage of Grafana
- Future work

Monitoring at RAL

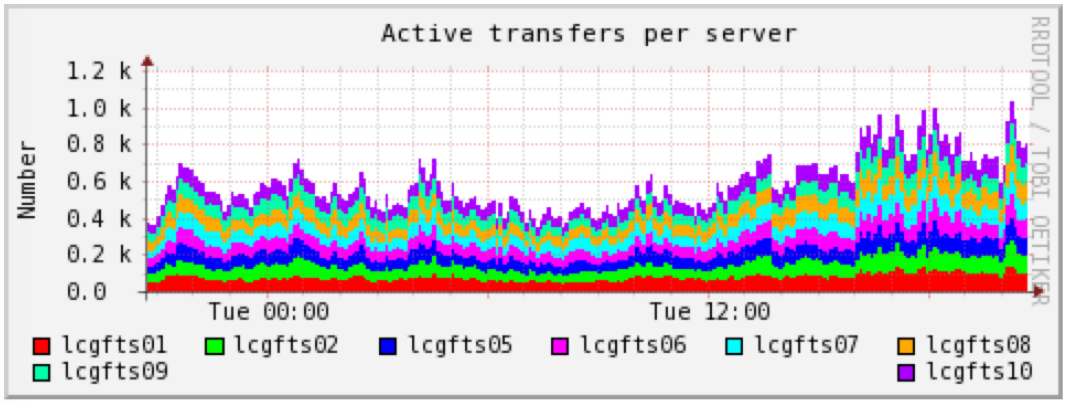
- Ganglia used at RAL
  - have ~ 89000 individual metrics
- Lots of problems
  - Plots don't look good
  - Difficult & time-consuming to make "nice" custom plots
    - we use Perl scripts, many are big, messy, complex, hard to maintain, generate hundreds of errors in httpd logs whenever someone looks at a plot
  - UI for custom plots is limited & makes bad plots anyway
  - gmond sometimes uses lots of memory & kills other things
  - doesn't handle dynamic resources well
  - not suitable for Ceph

A possible alternative

- InfluxDB + Grafana
  - InfluxDB is a time-series database
  - Grafana is a metrics dashboard
- Benefits
  - both are very easy to install
    - install rpm, then start the service
  - easy to put data into InfluxDB
  - easy to make nice plots in Grafana

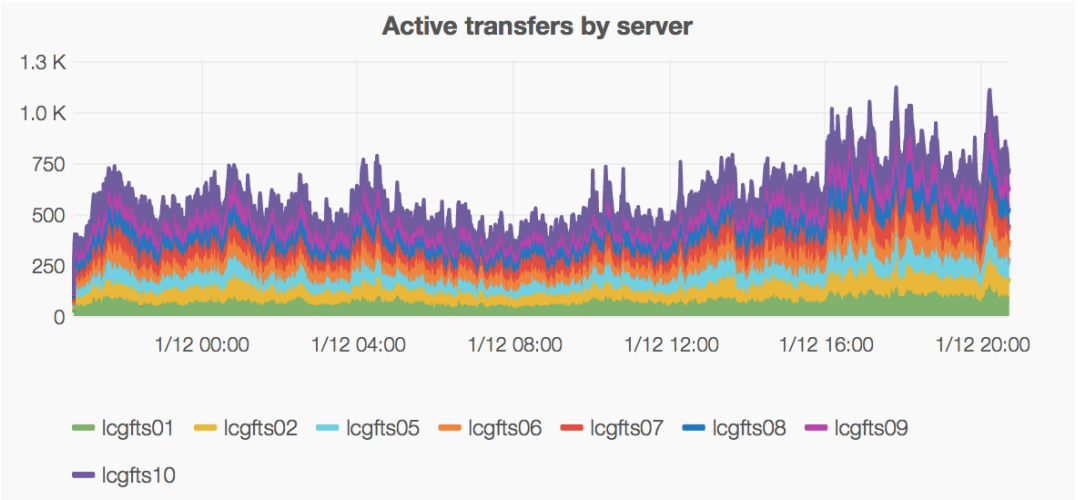
Monitoring at RAL

Go from



Ganglia

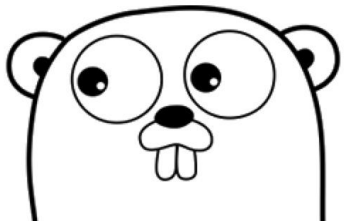
to



Grafana

InfluxDB

- Time series database
- Written in Go - no external depedencies
- SQL-like query language - InfluxQL
- Distributed (or not)
  - can be run as a single node
  - can be run as a cluster for redundancy & performance
    - will come back to this later
- Data can be written into InfluxDB in many ways
  - REST
  - API (e.g. Python)
  - Graphite, collectd



InfluxDB

- Data organized by time series, grouped together into databases
- Time series can have zero to many points
- Each point consists of
  - **time**
  - a **measurement**
    - e.g. cpu_load
  - at least one key-value **field**
    - e.g. value=5
  - zero to many **tags** containing metadata
    - e.g. host=lcg1423.gridpp.rl.ac.uk

InfluxDB

- Points written into InfluxDB using the line protocol format

```
<measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-key>=<field2-value>...] [timestamp]
```

- Example for an FTS3 server

```
active_transfers,host=lcgfts01,vo=atlas value=21
```

- Can write multiple points in batches to get better performance
  - this is recommended
  - example with 0.9.6.1-1 for 2000 points
    - sequentially: **129.7s**
    - in a batch: **0.16s**

Retention policies

- Retention policy describes
  - duration: how long data is kept
  - replication factor: how many copies of the data are kept
    - only for clusters ☺
- Can have multiple retention policies per database

Continuous queries

- An InfluxQL query that runs automatically & periodically within a database
- Mainly useful for downsampling data
  - read data from one retention policy
  - write downsampled data into another
- Example
  - database with 2 retention policies
    - 2 hour duration
    - keep forever
  - data with 1 second time resolution kept for 2 hours, data with 30 min time resolution kept forever
  - use a continuous query to aggregate the high time resolution data to 30 min time resolution

Example queries

```
> use arc
Using database arc

> show measurements
name: measurements
------------------
name
arex_heartbeat_lastseen
jobs
```

Example queries

```
> show tag keys from jobs
name: jobs
----------
tagKey
host
state
```

Example queries

```
> show tag values from jobs with key=host
name: hostTagValues
-------------------
host
arc-ce01
arc-ce02
arc-ce03
arc-ce04
```

Example queries

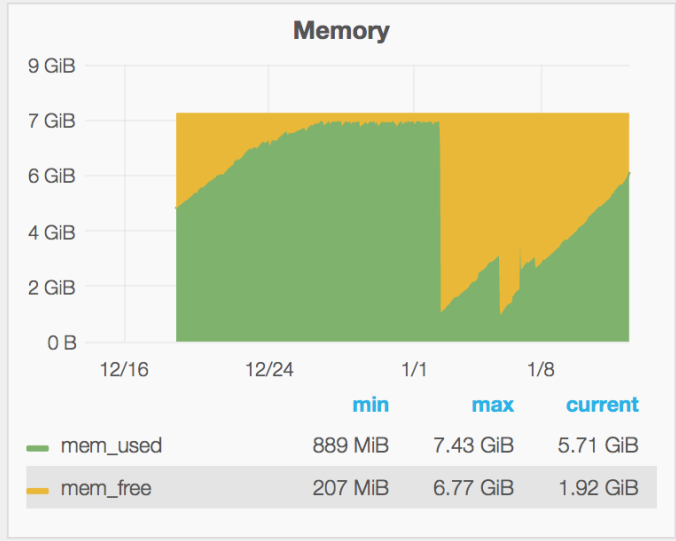
```
> select value,vo from active_transfers where
host='lcgfts01' and time > now() - 3m
name: active_transfers
----------------------
time                                value  vo
2016-01-14T21:25:02.143556502Z      100    cms
2016-01-14T21:25:02.143556502Z      7      cms/becms
2016-01-14T21:26:01.256006762Z      102    cms
2016-01-14T21:26:01.256006762Z      8      cms/becms
2016-01-14T21:27:01.455021342Z      97     cms
2016-01-14T21:27:01.455021342Z      7      cms/becms
2016-01-14T21:27:01.455021342Z      1      cms/dcms
```

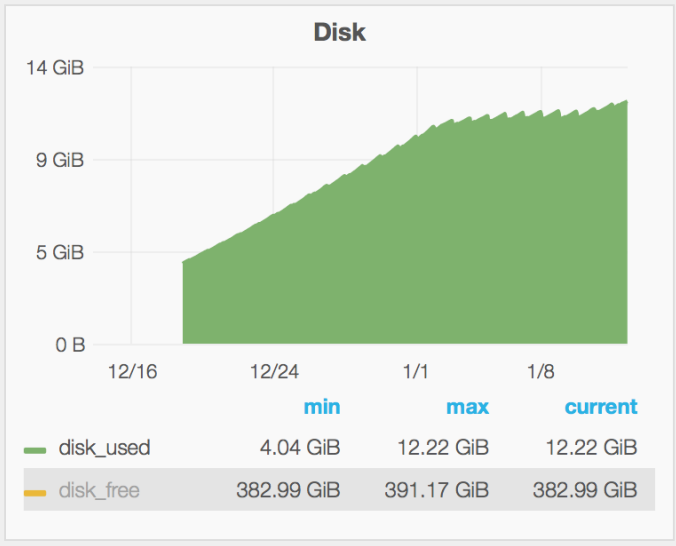
InfluxDB at RAL

- Single node instance
  - VM with 8 GB RAM, 4 cores
  - latest stable release of InfluxDB (0.9.6.1-1)
  - almost treated as a ‘production’ service
- What data is being sent to it?
  - Mainly application-specific metrics
  - Metrics from FTS3, HTCondor, ARC CEs, HAProxy, MariaDB, Mesos, OpenNebula, Windows Hypervisors, ...
- Cluster instance
  - currently just for testing
  - 6 bare-metal machines (ex worker nodes)
  - recent nightly build of InfluxDB

InfluxDB at RAL

- InfluxDB resource usage over the past month
  - currently using 1 month retention policies (1 min time resolution)
  - CPU usage negligible so far





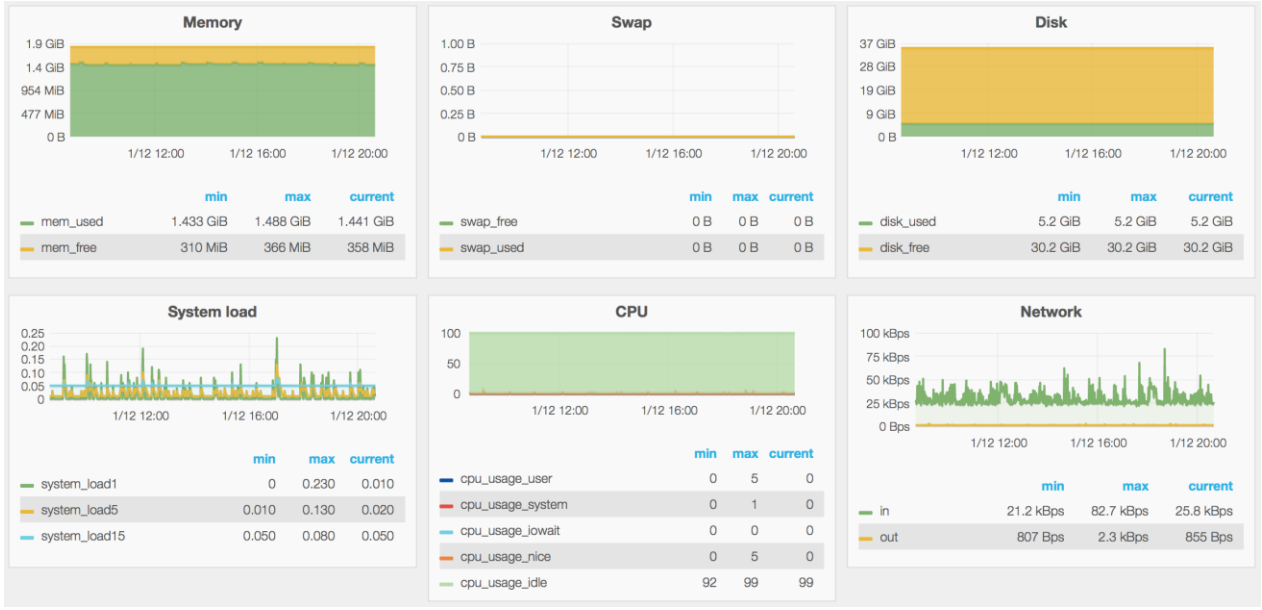
Sending metrics to InfluxDB

- Python scripts, using python-requests
- read InfluxDB host(s) from config file, for future cluster use
  - picks one at random, tries to write to it
  - if fails, picks another
  - ...
- Alternatively, can just use curl:

```
curl -s -X POST "http://<hostname>:8086/write?db=test" -u
user:passwd --data-binary "data,host=srv1 value=5"
```

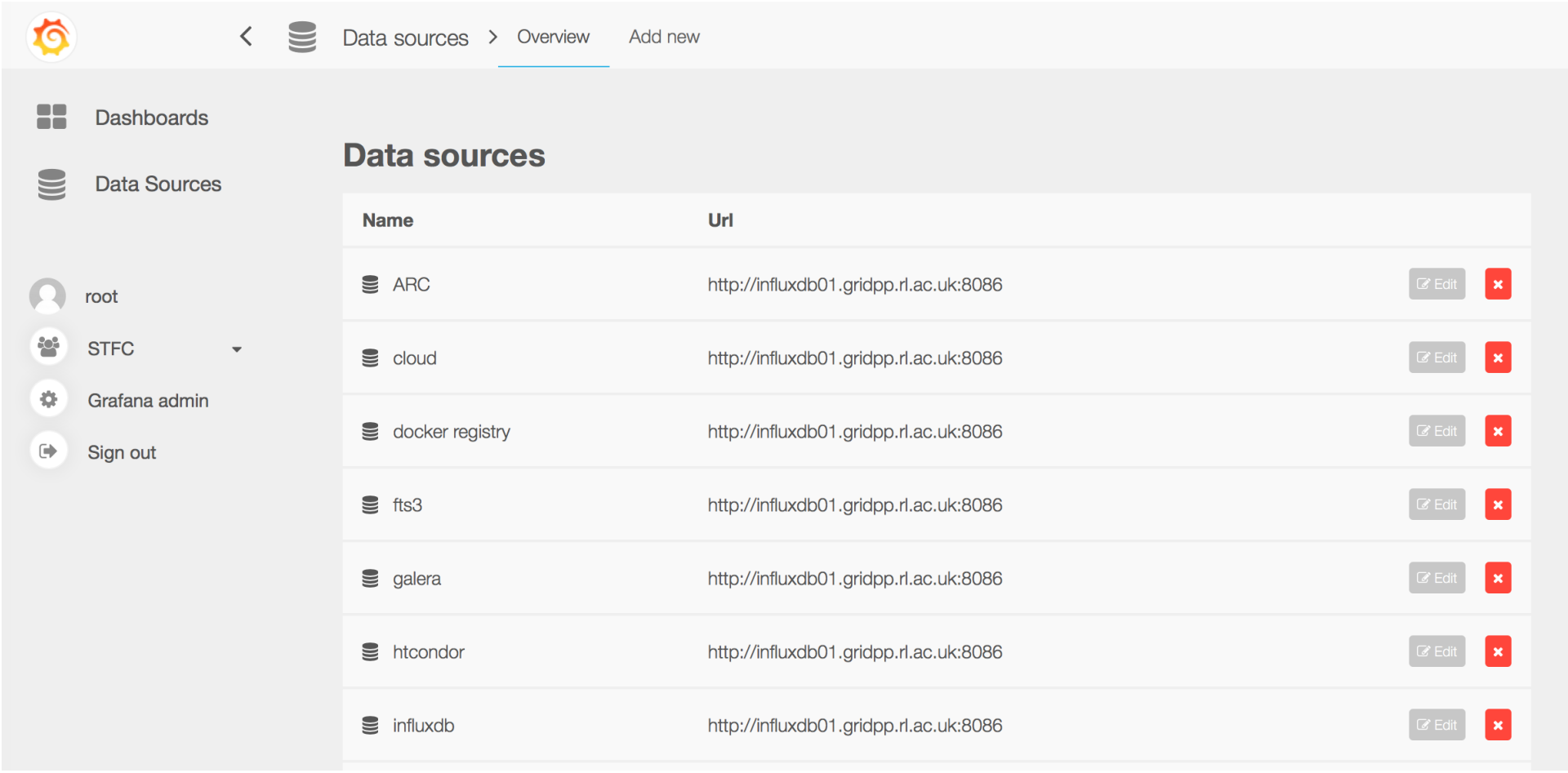
Telegraf

- Collects metrics & sends to InfluxDB
- Plugins for:
  - system (memory, load, CPU, network, disk, ...)
  - Apache, Elasticsearch, HAProxy, MySQL, Nginx, + many others

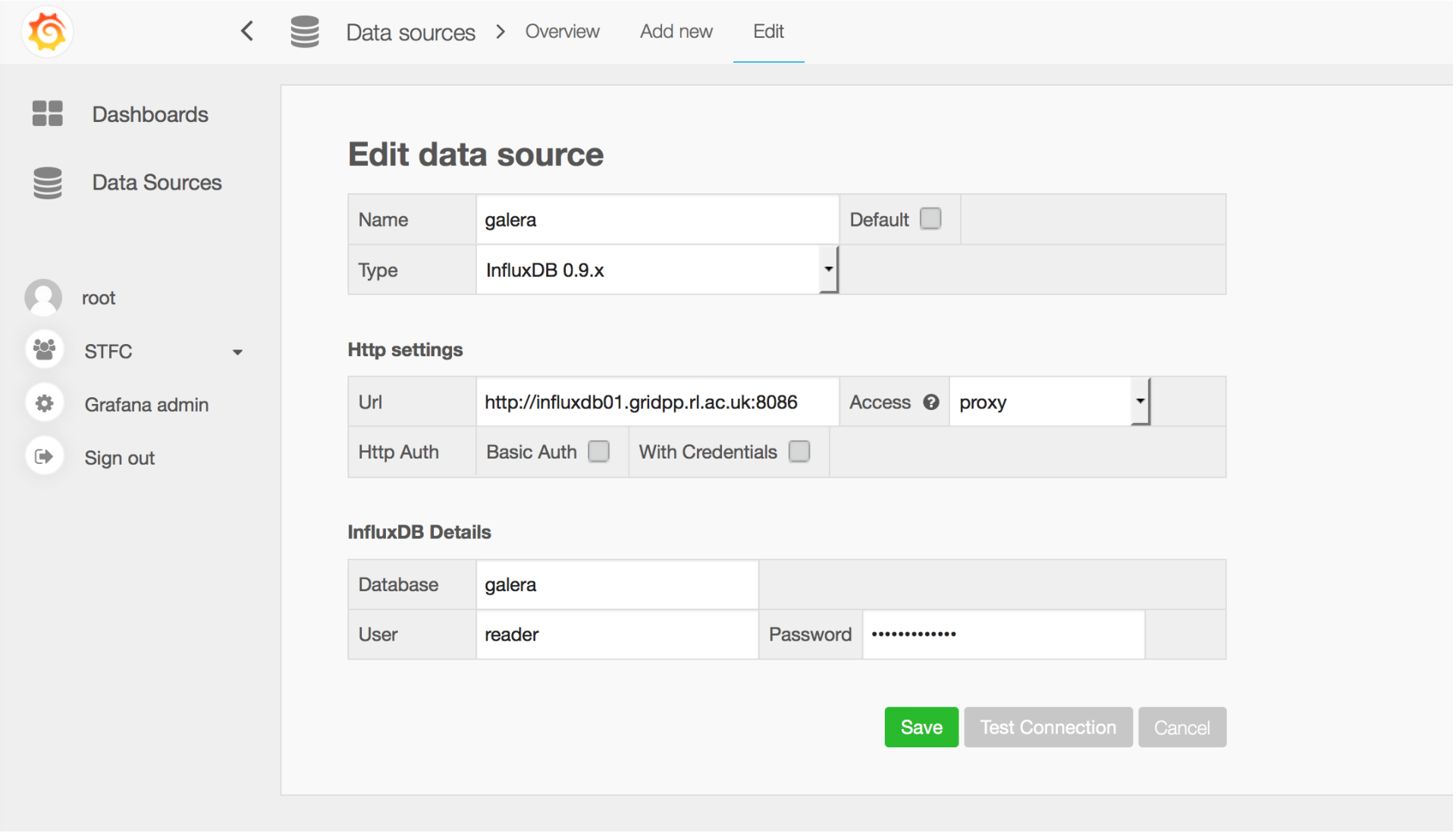


Example system metrics - Grafana

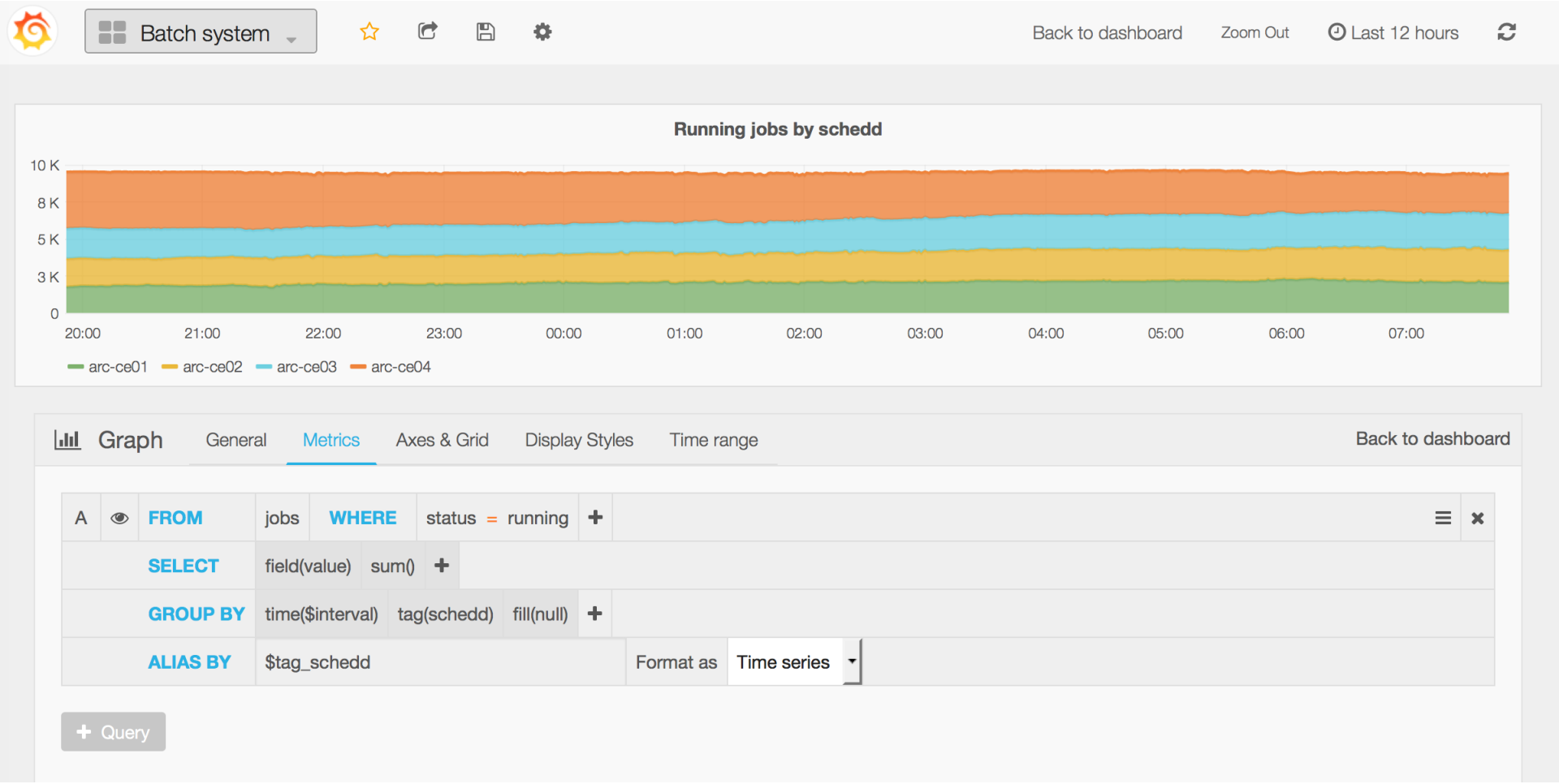
Grafana – data sources



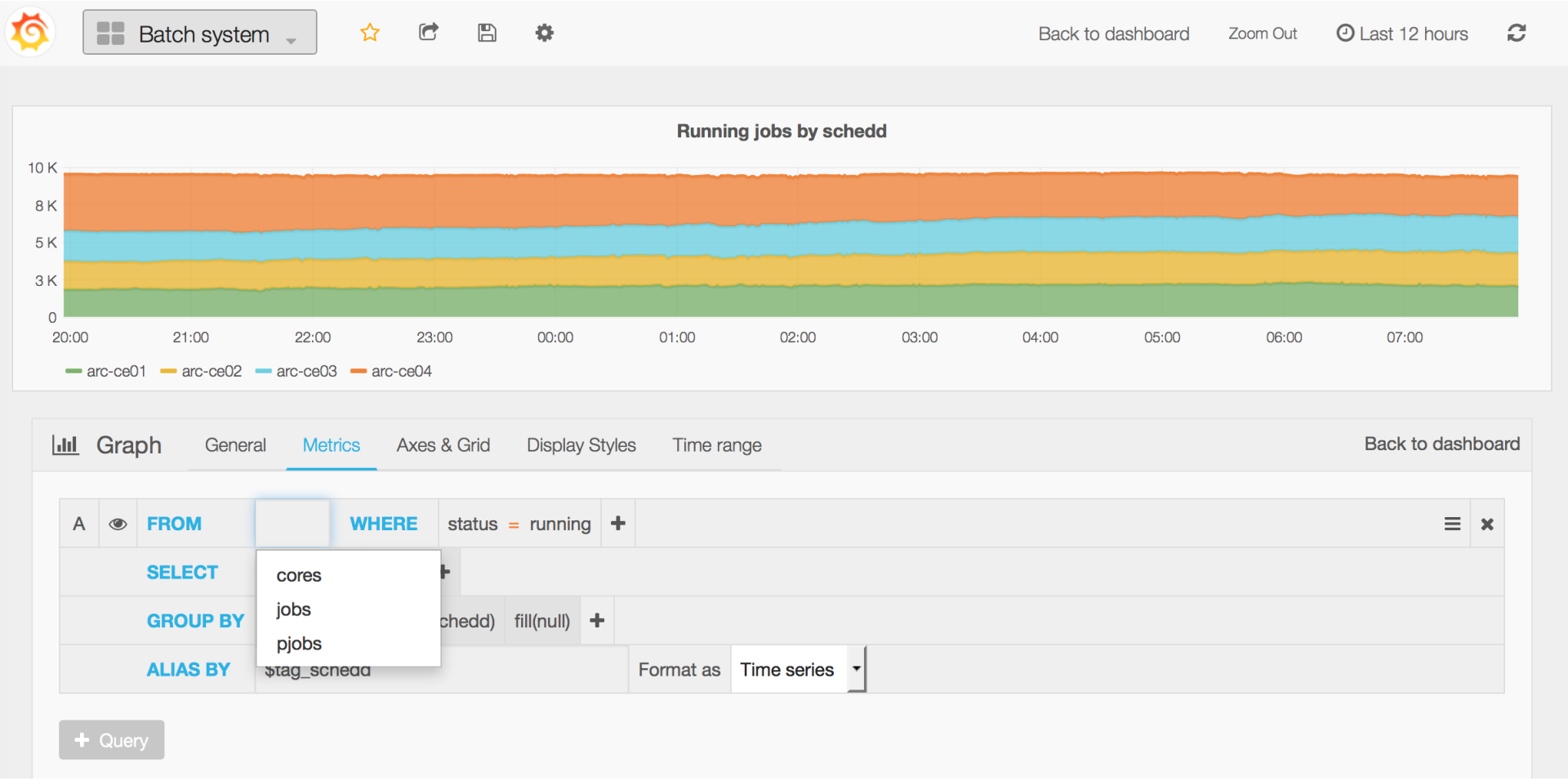
Grafana – adding a database



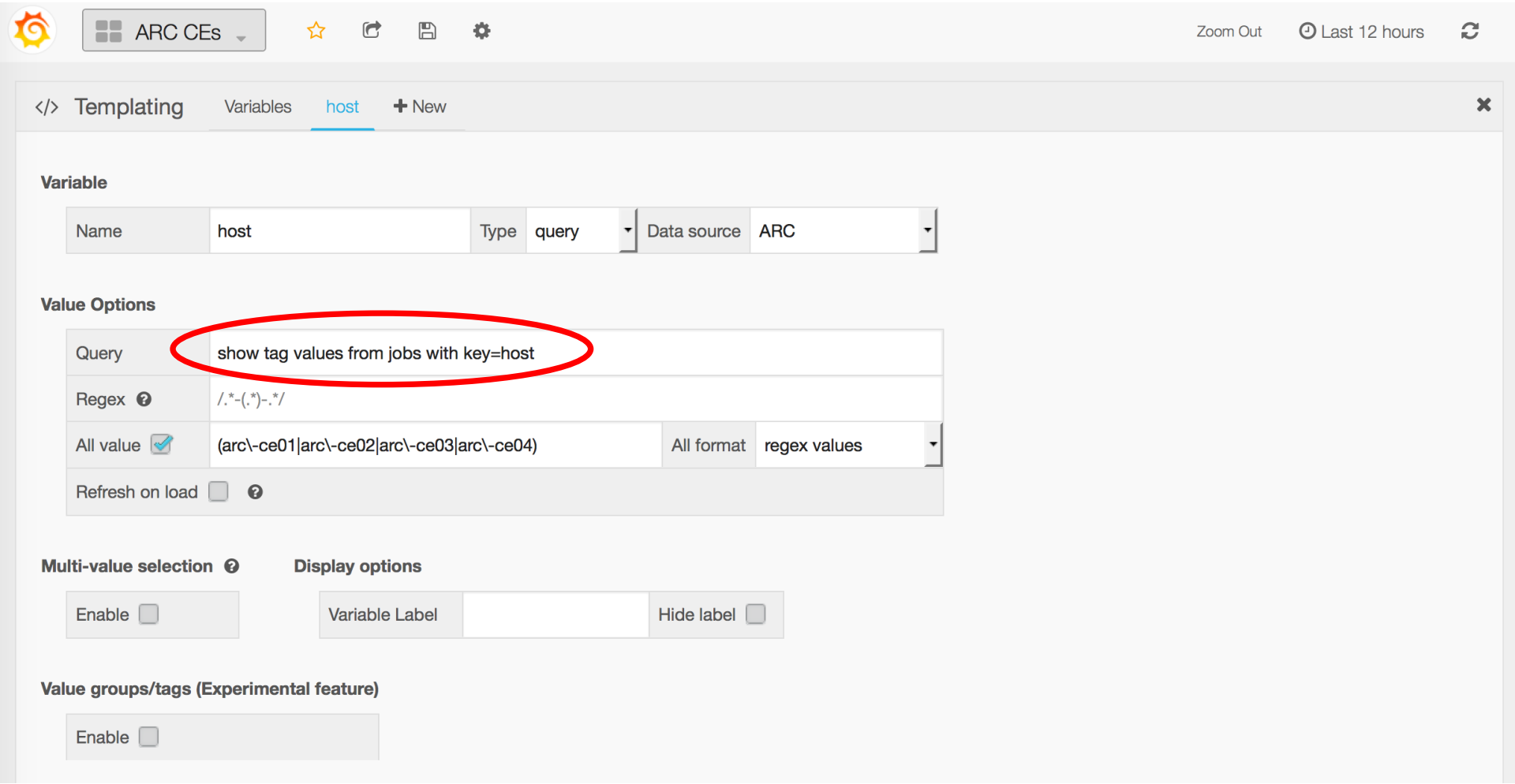
Grafana – making a plot



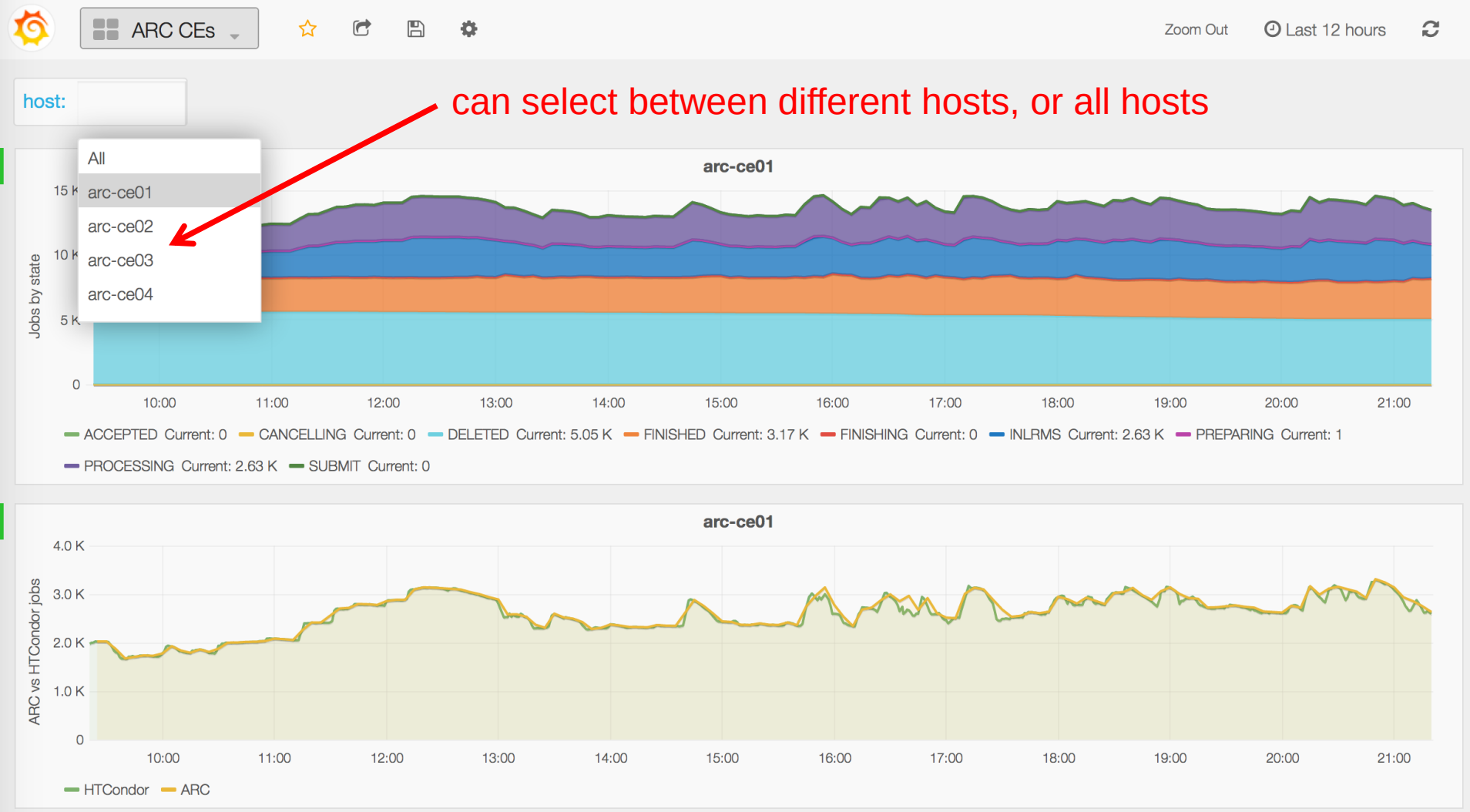
Grafana – making a plot



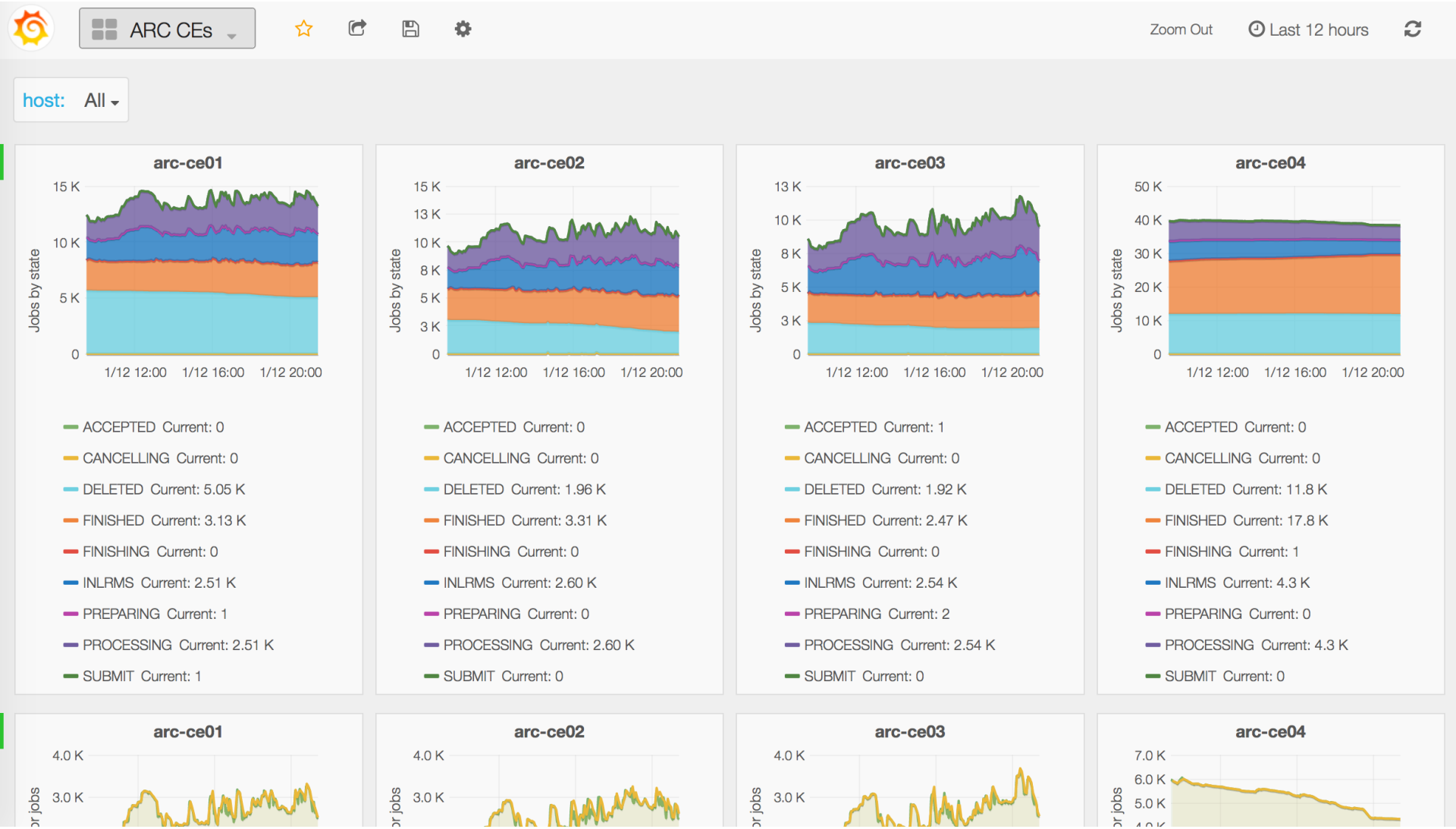
Templating


ARC CEs
Zoom Out
Last 12 hours
Templating
Variables
host
+ New
Variable
Name
host
Type
query
Data source
ARC
Value Options
Query
show tag values from jobs with key=host
Regex
/.*-(.*)-.*/
All value
(arc\-ce01|arc\-ce02|arc\-ce03|arc\-ce04)
All format
regex values
Refresh on load
Multi-value selection
Display options
Enable
Variable Label
Hide label
Value groups/tags (Experimental feature)
Enable

Templating

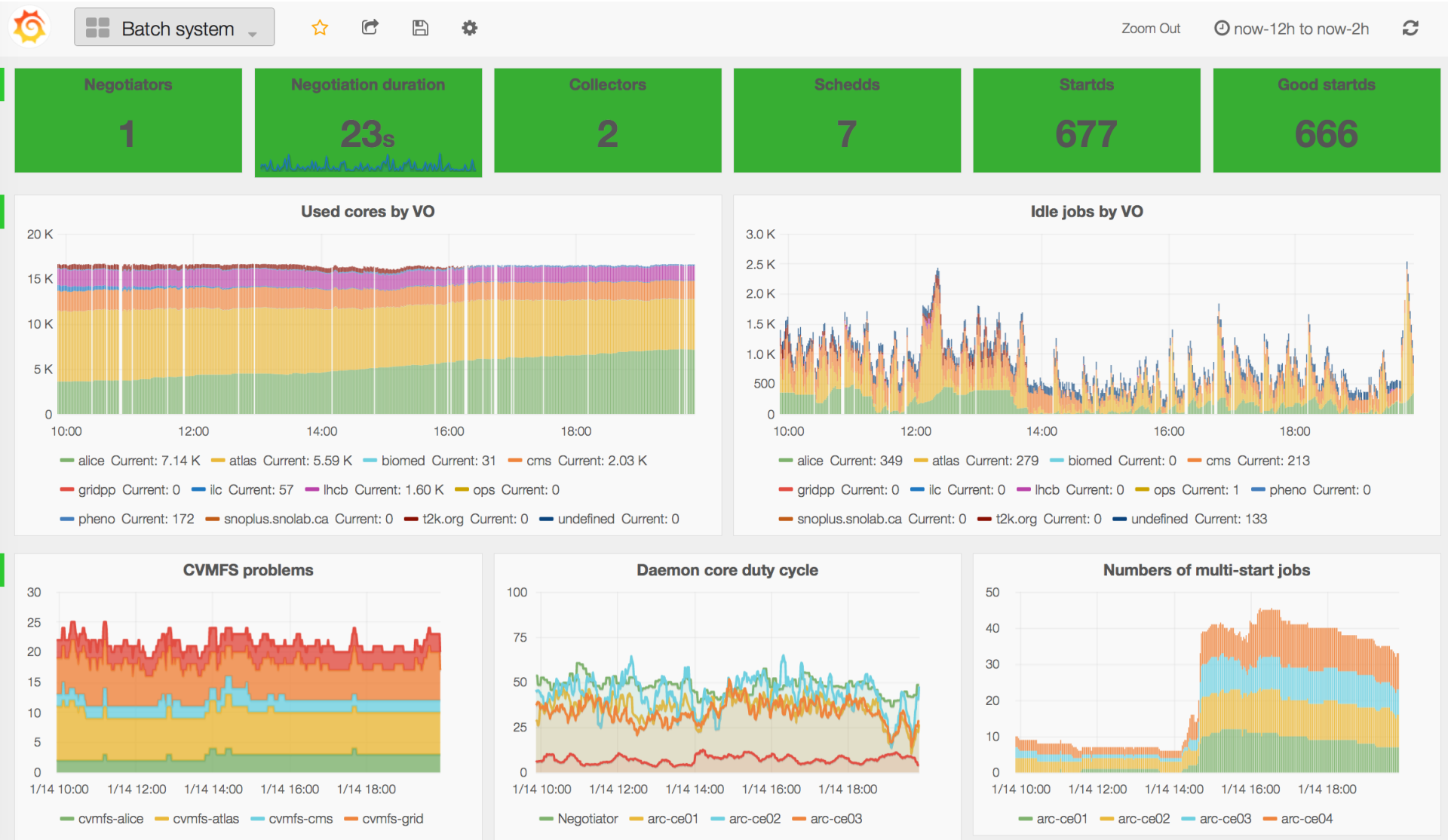

ARC CEs
Zoom Out
Last 12 hours
host:
can select between different hosts, or all hosts
All
arc-ce01
arc-ce02
arc-ce03
arc-ce04
arc-ce01
Jobs by state
15 K
10 K
5 K
0
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
ACCEPTED Current: 0
CANCELLING Current: 0
DELETED Current: 5.05 K
FINISHED Current: 3.17 K
FINISHING Current: 0
INLRMS Current: 2.63 K
PREPARING Current: 1
PROCESSING Current: 2.63 K
SUBMIT Current: 0
arc-ce01
ARC vs HTCondor jobs
4.0 K
3.0 K
2.0 K
1.0 K
0
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
HTCondor
ARC

Templating

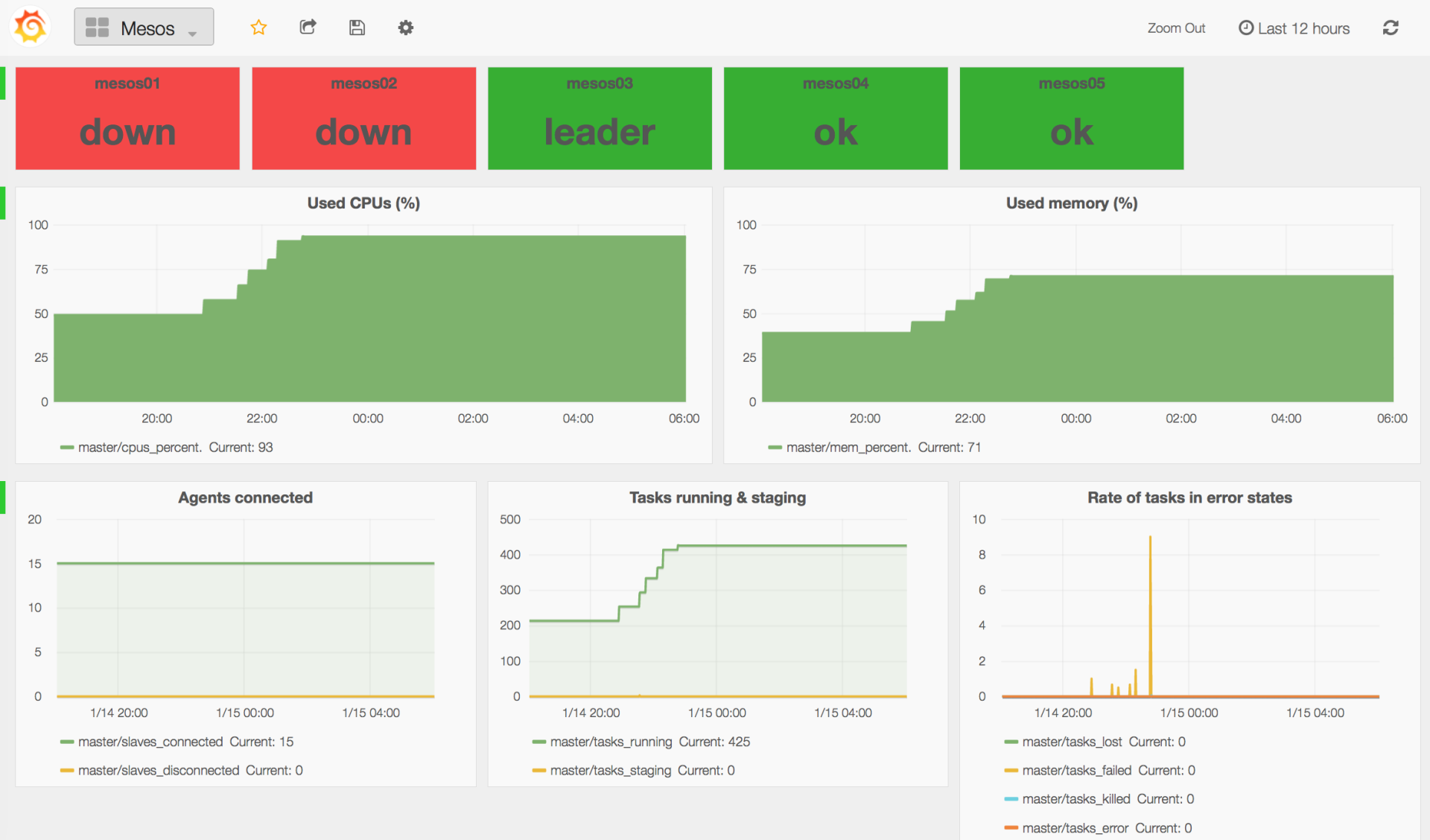

ARC CEs
Zoom Out
Last 12 hours
host: All
arc-ce01
Jobs by state
15 K
10 K
5 K
0
1/12 12:00
1/12 16:00
1/12 20:00
ACCEPTED Current: 0
CANCELLING Current: 0
DELETED Current: 5.05 K
FINISHED Current: 3.13 K
FINISHING Current: 0
INLRMS Current: 2.51 K
PREPARING Current: 1
PROCESSING Current: 2.51 K
SUBMIT Current: 1
arc-ce02
Jobs by state
15 K
13 K
10 K
8 K
5 K
3 K
0
1/12 12:00
1/12 16:00
1/12 20:00
ACCEPTED Current: 0
CANCELLING Current: 0
DELETED Current: 1.96 K
FINISHED Current: 3.31 K
FINISHING Current: 0
INLRMS Current: 2.60 K
PREPARING Current: 0
PROCESSING Current: 2.60 K
SUBMIT Current: 0
arc-ce03
Jobs by state
13 K
10 K
8 K
5 K
3 K
0
1/12 12:00
1/12 16:00
1/12 20:00
ACCEPTED Current: 1
CANCELLING Current: 0
DELETED Current: 1.92 K
FINISHED Current: 2.47 K
FINISHING Current: 0
INLRMS Current: 2.54 K
PREPARING Current: 2
PROCESSING Current: 2.54 K
SUBMIT Current: 0
arc-ce04
Jobs by state
50 K
40 K
30 K
20 K
10 K
0
1/12 12:00
1/12 16:00
1/12 20:00
ACCEPTED Current: 0
CANCELLING Current: 0
DELETED Current: 11.8 K
FINISHED Current: 17.8 K
FINISHING Current: 1
INLRMS Current: 4.3 K
PREPARING Current: 0
PROCESSING Current: 4.3 K
SUBMIT Current: 0
arc-ce01
4.0 K
3.0 K
arc-ce02
4.0 K
3.0 K
arc-ce03
4.0 K
3.0 K
arc-ce04
7.0 K
6.0 K
5.0 K

Example dashboards

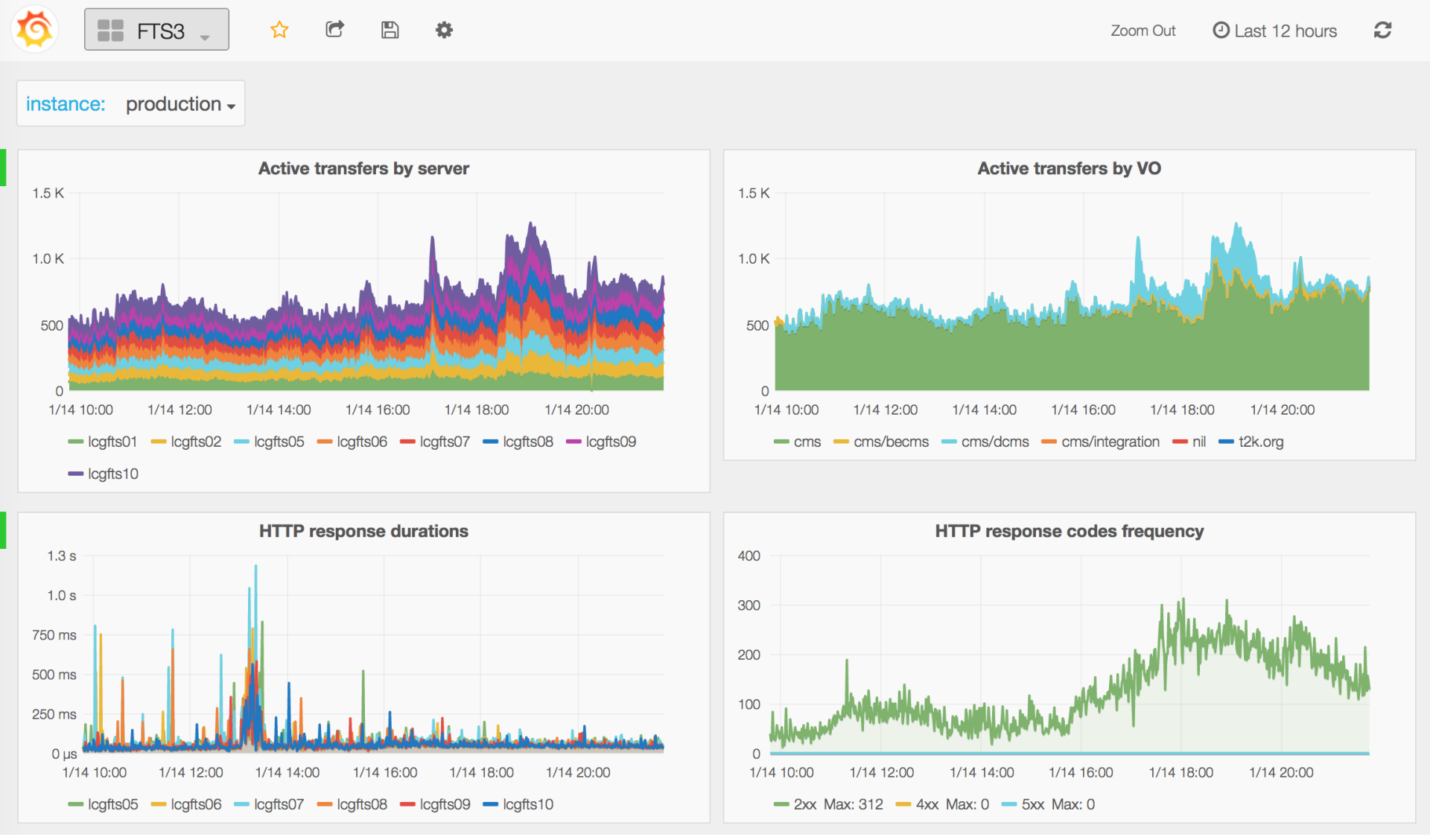
HTCondor


Batch system
Zoom Out
now-12h to now-2h
Negotiators
1
Negotiation duration
23s
Collectors
2
Schedds
7
Startds
677
Good startds
666
Used cores by VO
20 K
15 K
10 K
5 K
0
10:00
12:00
14:00
16:00
18:00
alice Current: 7.14 K
atlas Current: 5.59 K
biomed Current: 31
cms Current: 2.03 K
gridpp Current: 0
ilc Current: 57
lhcb Current: 1.60 K
ops Current: 0
pheno Current: 172
snoplus.snolab.ca Current: 0
t2k.org Current: 0
undefined Current: 0
Idle jobs by VO
3.0 K
2.5 K
2.0 K
1.5 K
1.0 K
500
0
10:00
12:00
14:00
16:00
18:00
alice Current: 349
atlas Current: 279
biomed Current: 0
cms Current: 213
gridpp Current: 0
ilc Current: 0
lhcb Current: 0
ops Current: 1
pheno Current: 0
snoplus.snolab.ca Current: 0
t2k.org Current: 0
undefined Current: 133
CVMFS problems
30
25
20
15
10
5
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
cvmfs-alice
cvmfs-atlas
cvmfs-cms
cvmfs-grid
Daemon core duty cycle
100
75
50
25
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
Negotiator
arc-ce01
arc-ce02
arc-ce03
Numbers of multi-start jobs
50
40
30
20
10
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
arc-ce01
arc-ce02
arc-ce03
arc-ce04

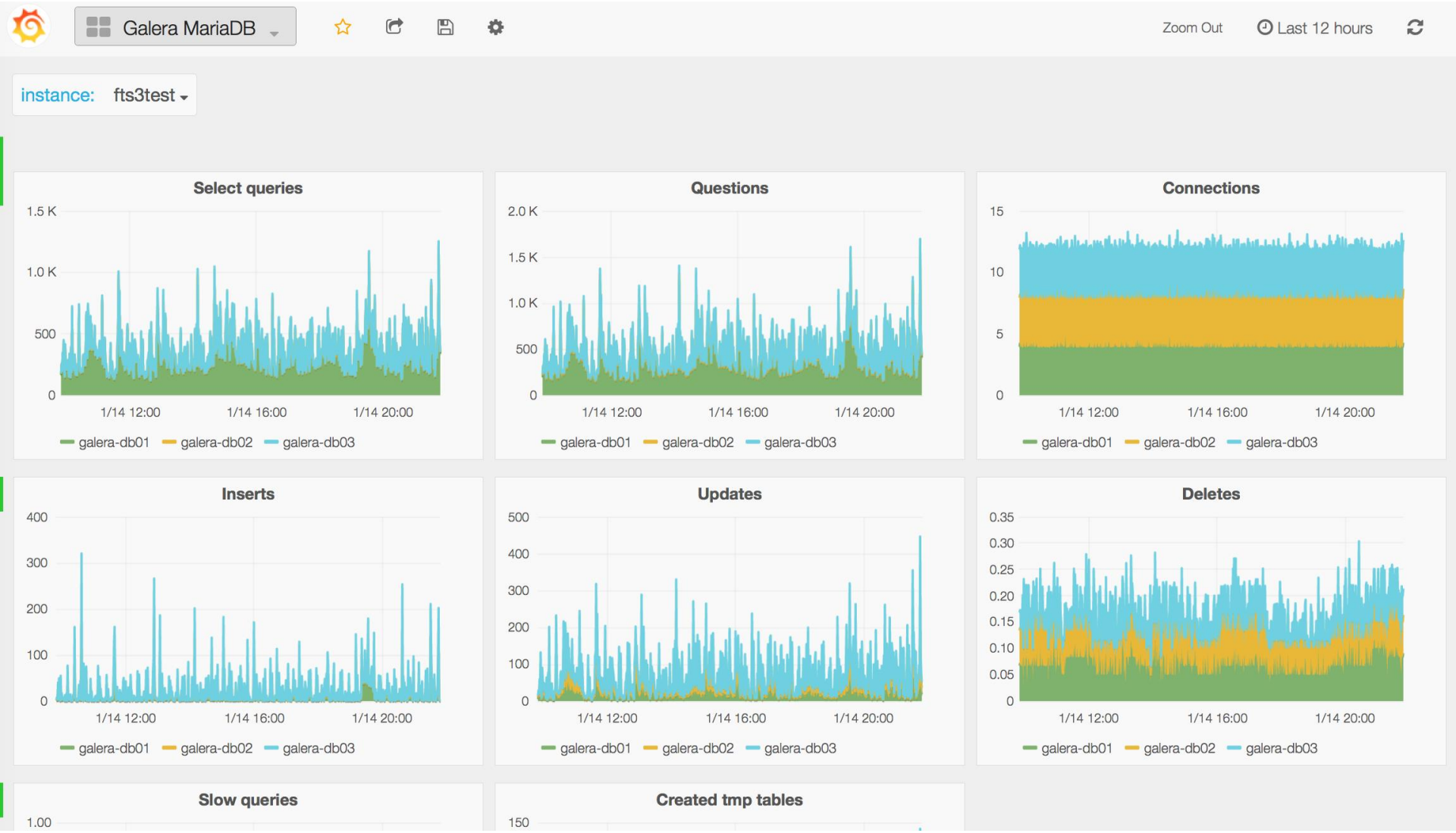
Mesos


Mesos
Zoom Out
Last 12 hours
mesos01
down
mesos02
down
mesos03
leader
mesos04
ok
mesos05
ok
Used CPUs (%)
100
75
50
25
0
20:00
22:00
00:00
02:00
04:00
06:00
master/cpus_percent. Current: 93
Used memory (%)
100
75
50
25
0
20:00
22:00
00:00
02:00
04:00
06:00
master/mem_percent. Current: 71
Agents connected
20
15
10
5
0
1/14 20:00
1/15 00:00
1/15 04:00
master/slaves_connected Current: 15
master/slaves_disconnected Current: 0
Tasks running & staging
500
400
300
200
100
0
1/14 20:00
1/15 00:00
1/15 04:00
master/tasks_running Current: 425
master/tasks_staging Current: 0
Rate of tasks in error states
10
8
6
4
2
0
1/14 20:00
1/15 00:00
1/15 04:00
master/tasks_lost Current: 0
master/tasks_failed Current: 0
master/tasks_killed Current: 0
master/tasks_error Current: 0

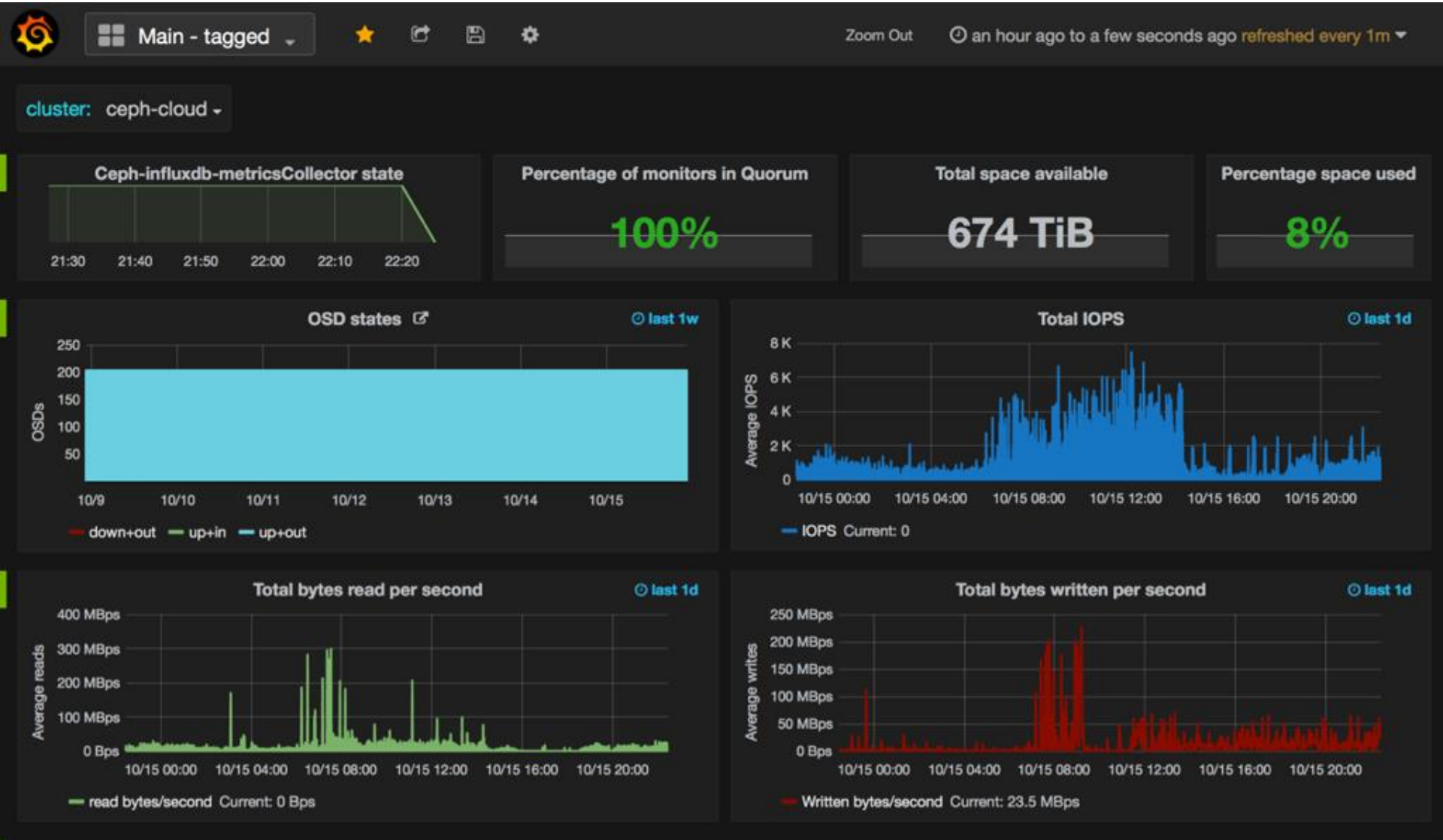
FTS3


FTS3
Zoom Out
Last 12 hours
instance: production
Active transfers by server
1.5 K
1.0 K
500
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
1/14 20:00
lcgfts01 lcgfts02 lcgfts05 lcgfts06 lcgfts07 lcgfts08 lcgfts09
lcgfts10
Active transfers by VO
1.5 K
1.0 K
500
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
1/14 20:00
cms cms/becms cms/dcms cms/integration nil t2k.org
HTTP response durations
1.3 s
1.0 s
750 ms
500 ms
250 ms
0 µs
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
1/14 20:00
lcgfts05 lcgfts06 lcgfts07 lcgfts08 lcgfts09 lcgfts10
HTTP response codes frequency
400
300
200
100
0
1/14 10:00
1/14 12:00
1/14 14:00
1/14 16:00
1/14 18:00
1/14 20:00
2xx Max: 312 4xx Max: 0 5xx Max: 0

Databases


Galera MariaDB
Zoom Out
Last 12 hours
instance: fts3test
Select queries
1.5 K
1.0 K
500
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Questions
2.0 K
1.5 K
1.0 K
500
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Connections
15
10
5
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Inserts
400
300
200
100
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Updates
500
400
300
200
100
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Deletes
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
1/14 12:00
1/14 16:00
1/14 20:00
galera-db01
galera-db02
galera-db03
Slow queries
1.00
Created tmp tables
150

Ceph


Main - tagged
Zoom Out
an hour ago to a few seconds ago refreshed every 1m
cluster: ceph-cloud
Ceph-influxdb-metricsCollector state
21:30 21:40 21:50 22:00 22:10 22:20
Percentage of monitors in Quorum
100%
Total space available
674 TiB
Percentage space used
8%
OSD states
last 1w
OSDs
250 200 150 100 50
10/9 10/10 10/11 10/12 10/13 10/14 10/15
down+out up+in up+out
Total IOPS
last 1d
Average IOPS
8 K 6 K 4 K 2 K 0
10/15 00:00 10/15 04:00 10/15 08:00 10/15 12:00 10/15 16:00 10/15 20:00
IOPS Current: 0
Total bytes read per second
last 1d
Average reads
400 MBps 300 MBps 200 MBps 100 MBps 0 Bps
10/15 00:00 10/15 04:00 10/15 08:00 10/15 12:00 10/15 16:00 10/15 20:00
read bytes/second Current: 0 Bps
Total bytes written per second
last 1d
Average writes
250 MBps 200 MBps 150 MBps 100 MBps 50 MBps 0 Bps
10/15 00:00 10/15 04:00 10/15 08:00 10/15 12:00 10/15 16:00 10/15 20:00
Written bytes/second Current: 23.5 MBps

Load testing InfluxDB

- Can a single InfluxDB node handle large numbers Telegraf instances sending data to it?
  - Telegraf configured to measure load, CPU, memory, swap, disk
  - testing done the night before my HEPiX Fall 2015 talk
    - 189 instances sending data each minute to InfluxDB 0.9.4

      had problems
  - testing yesterday
    - 412 instances sending data each minute to InfluxDB 0.9.6.1-1

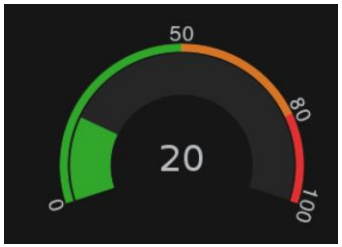
      no problems
    - couldn't try more – ran out of resources & couldn't create any more Telegraf containers

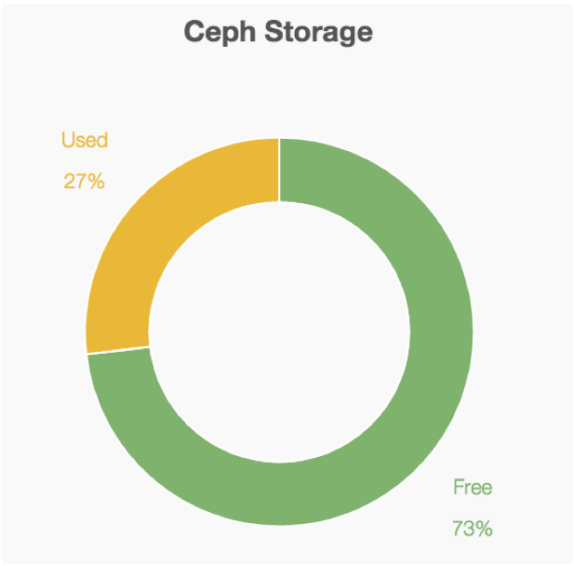
Current limitations

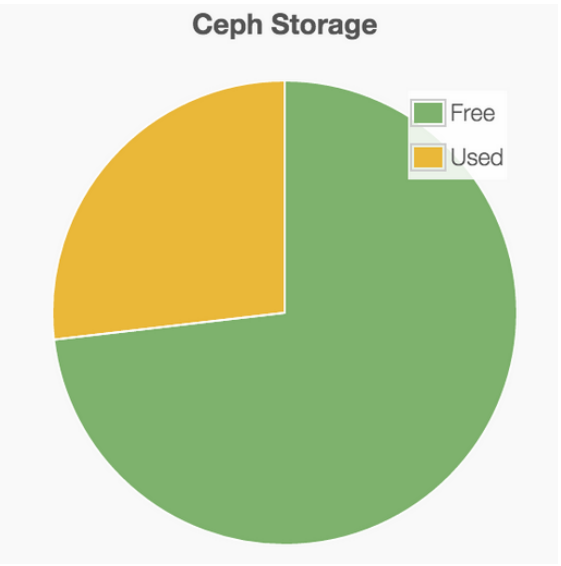
- (Grafana) long duration plots can be quite slow
  - e.g. 1 month plot, using 1-min resolution data
  - Possible fix: people have requested that Grafana should be able to automatically select different retention policies depending on time interval
- (InfluxDB) No way to automatically downsample all measurements in a database
  - need to have a continuous query per measurement
  - Possible fix: people have requested that it should be possible to use regular expressions in continuous queries

Upcoming features

- Grafana – gauges & pie charts in progress







Future work

- Re-test clustering once it becomes stable/fully-functional
  - expected to be available in 0.10 at end of January
  - also new storage engine, query engine, ...
- Investigate Kapacitor
  - time-series data processing engine, real-time or batch
  - trigger events/alerts, or send processed data back to InfluxDB
  - anomoly detection from service metrics