Python at cloudkick

Tomaz Muraus
tomaz.muraus@rackspace.com

June 22, 2011

EUROPYTHON 2011
FLORENCE, JUNE 20–26

rackspace

Agenda

- Who am I
- What is Cloudkick
- Cloudkick Architecture
- Why Python?
- Python at Cloudkick

Agenda

- Web Application & Django
- Backend / Network Services & Twisted
- Service Communication & RPC
- Testing
- Other places
- Questions

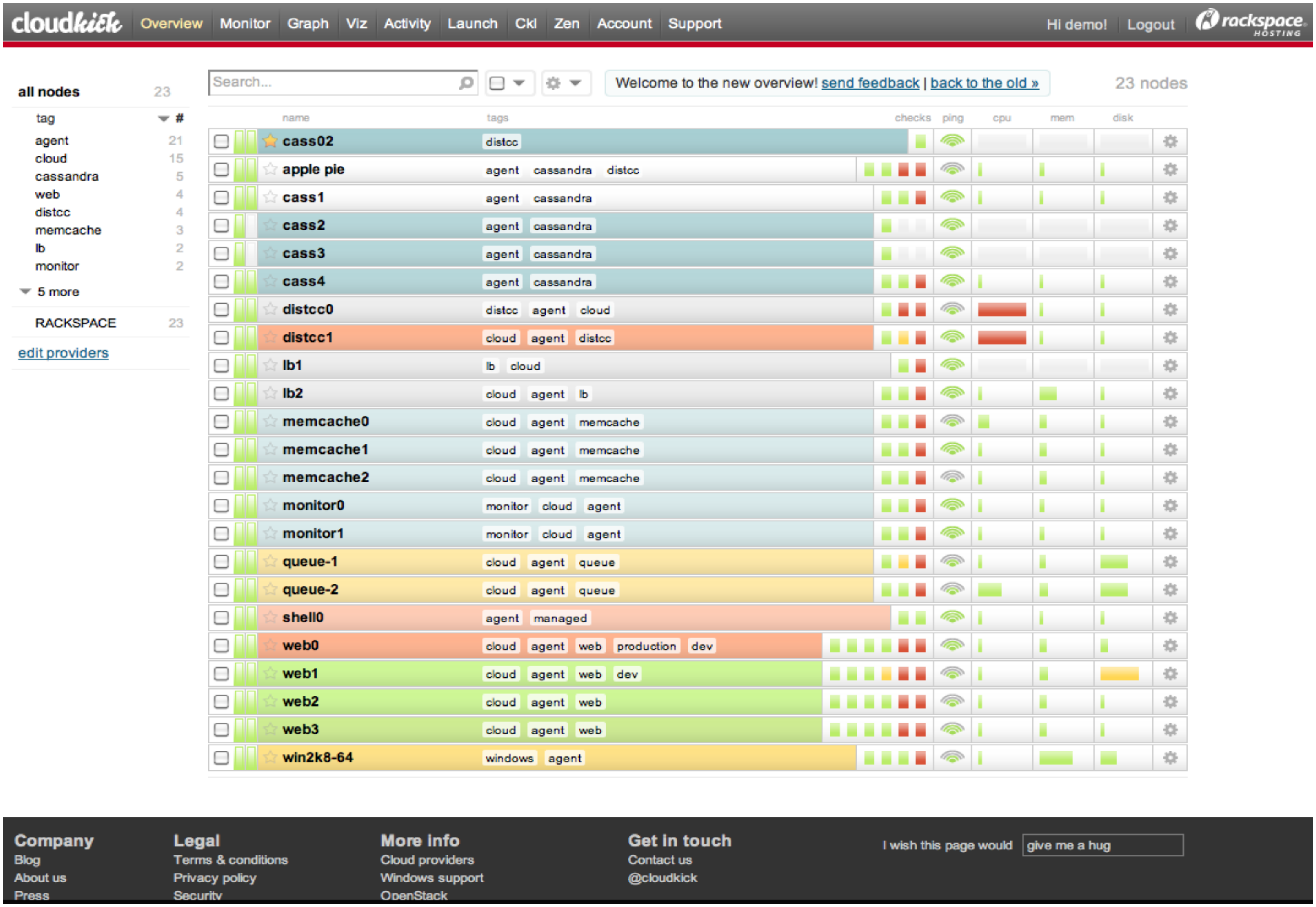
Who am I

- Tomaz Muraus, @KamiSLO
- Software developer at Cloudkick / Rackspace
- Author of multiple Python libraries and Django apps – http://github.com/Kami
- FOSS supporter & lover
- Apache Libcloud Committer

What is Cloudkick

- Server management and monitoring SaaS
- Manage all your cloud and physical servers from a single control panel
- Set up monitors and alerts – only get waken up when the stuff actually breaks
- Annotate graphs
- and more...

What is Cloudkick



Welcome to the new overview! send feedback | back to the old »

23 nodes

all nodes	23
tag	#
agent	21
cloud	15
cassandra	5
web	4
distcc	4
memcache	3
lb	2
monitor	2
5 more	
RACKSPACE	23

edit providers

name	tags	checks
cass02	distcc	🟩
apple pie	agent cassandra distcc	🟩🟩🟥🟥
cass1	agent cassandra	🟩🟩🟥
cass2	agent cassandra	🟩
cass3	agent cassandra	🟩
cass4	agent cassandra	🟩🟩🟥
distcc0	distcc agent cloud	🟩🟥🟥
distcc1	cloud agent distcc	🟩🟨🟥
lb1	lb cloud	🟩🟥
lb2	cloud agent lb	🟩🟩🟥
memcache0	cloud agent memcache	🟩🟩🟥
memcache1	cloud agent memcache	🟩🟩🟥
memcache2	cloud agent memcache	🟩🟩🟥
monitor0	monitor cloud agent	🟩🟩🟥
monitor1	monitor cloud agent	🟩🟩🟥
queue-1	cloud agent queue	🟩🟨🟥
queue-2	cloud agent queue	🟩🟩🟥
shell0	agent managed	🟩🟩
web0	cloud agent web production dev	🟩🟩🟩🟩🟥🟥
web1	cloud agent web dev	🟩🟩🟩🟨🟥🟥
web2	cloud agent web	🟩🟩🟩🟩🟥🟥
web3	cloud agent web	🟩🟩🟩🟩🟥🟥
win2k8-64	windows agent	🟩🟩🟩🟥

Company
Blog
About us
Press

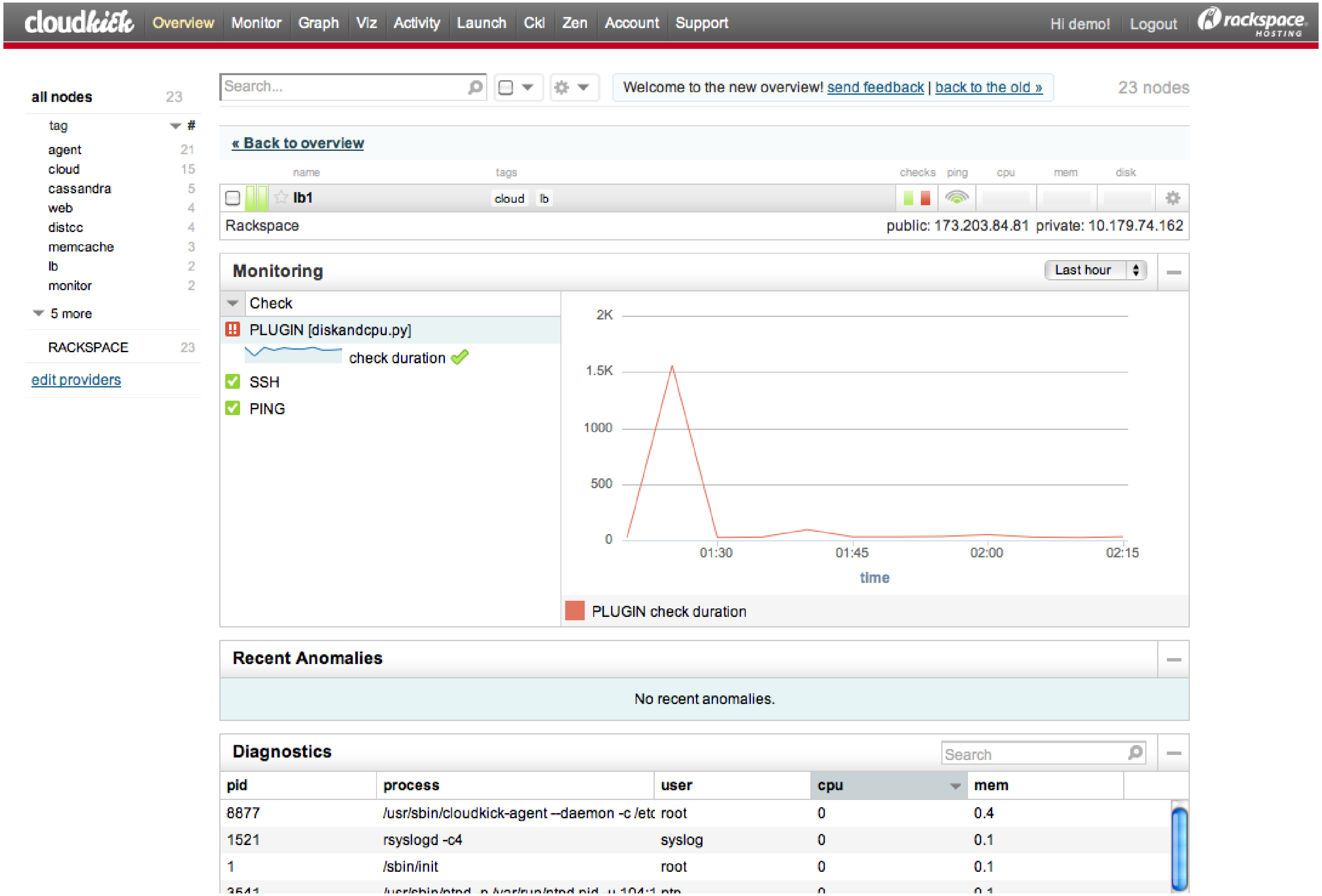
Legal
Terms & conditions
Privacy policy
Security

More info
Cloud providers
Windows support
OpenStack

Get in touch
Contact us
@cloudkick

I wish this page would give me a hug

What is Cloudkick



all nodes	23
tag	#
agent	21
cloud	15
cassandra	5
web	4
distcc	4
memcache	3
lb	2
monitor	2
5 more	
RACKSPACE	23

edit providers

Welcome to the new overview! send feedback | back to the old »

23 nodes

« Back to overview

□	name	tags	checks	ping	cpu	mem	disk
□	☆ lb1	cloud lb					

Rackspace

public: 173.203.84.81 private: 10.179.74.162

Monitoring

Last hour

Check

- PLUGIN [diskandcpu.py]
  - check duration ✓
- ☑ SSH
- ☑ PING

🟥 PLUGIN check duration

Recent Anomalies

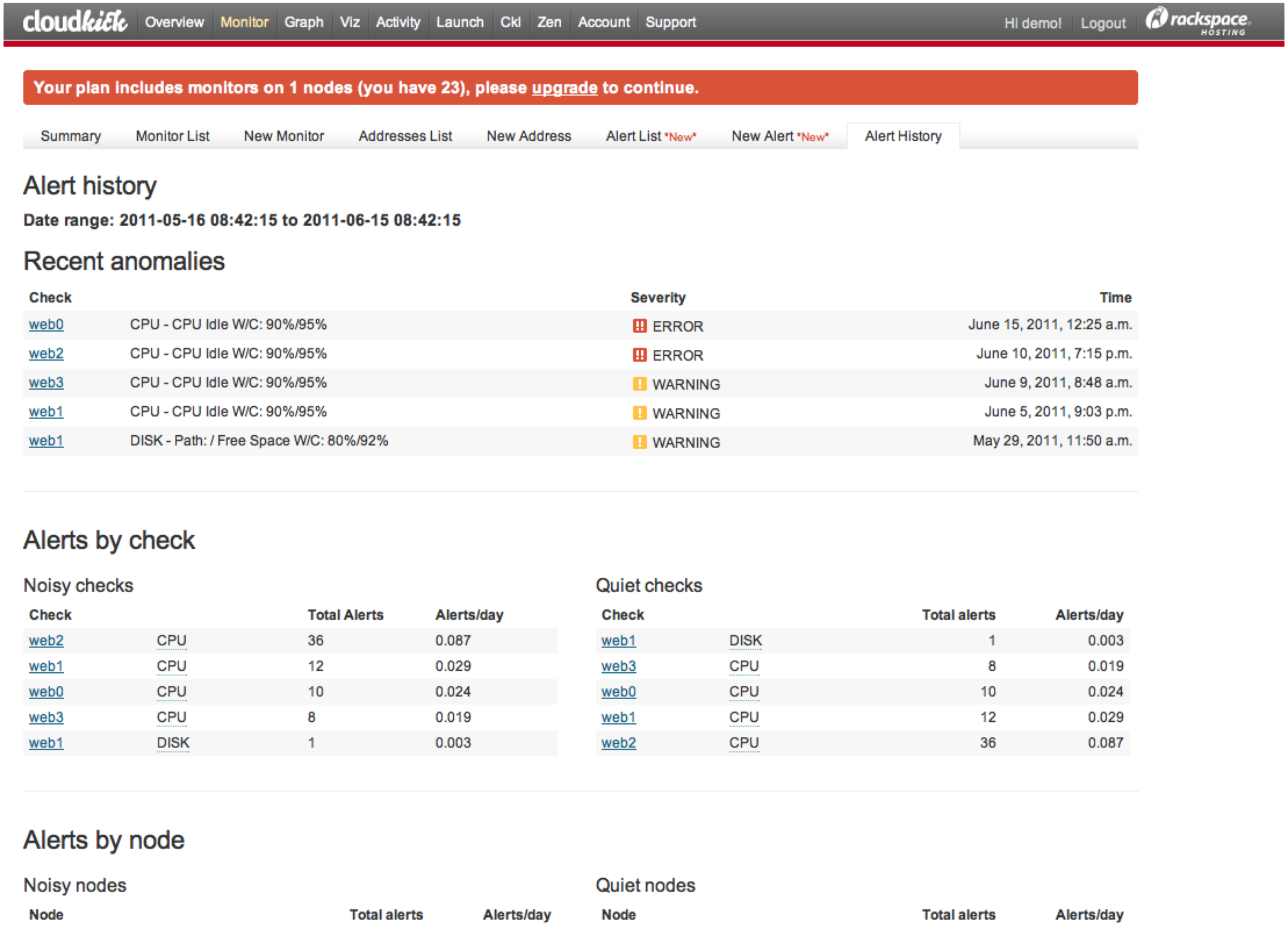
No recent anomalies.

Diagnostics

Search

pid	process	user	cpu	mem
8877	/usr/sbin/cloudkick-agent --daemon -c /etc	root	0	0.4
1521	rsyslogd -c4	syslog	0	0.1
1	/sbin/init	root	0	0.1
3541	/usr/sbin/ntpd -p /var/run/ntpd.pid -u 104:1	ntp	0	0.1

What is Cloudkick



Your plan includes monitors on 1 nodes (you have 23), please upgrade to continue.

Alert history

Date range: 2011-05-16 08:42:15 to 2011-06-15 08:42:15

Recent anomalies

Check		Severity	Time
web0	CPU - CPU Idle W/C: 90%/95%	‼ ERROR	June 15, 2011, 12:25 a.m.
web2	CPU - CPU Idle W/C: 90%/95%	‼ ERROR	June 10, 2011, 7:15 p.m.
web3	CPU - CPU Idle W/C: 90%/95%	! WARNING	June 9, 2011, 8:48 a.m.
web1	CPU - CPU Idle W/C: 90%/95%	! WARNING	June 5, 2011, 9:03 p.m.
web1	DISK - Path: / Free Space W/C: 80%/92%	! WARNING	May 29, 2011, 11:50 a.m.

Alerts by check

Noisy checks

Check		Total Alerts	Alerts/day
web2	CPU	36	0.087
web1	CPU	12	0.029
web0	CPU	10	0.024
web3	CPU	8	0.019
web1	DISK	1	0.003

Quiet checks

Check		Total alerts	Alerts/day
web1	DISK	1	0.003
web3	CPU	8	0.019
web0	CPU	10	0.024
web1	CPU	12	0.029
web2	CPU	36	0.087

Alerts by node

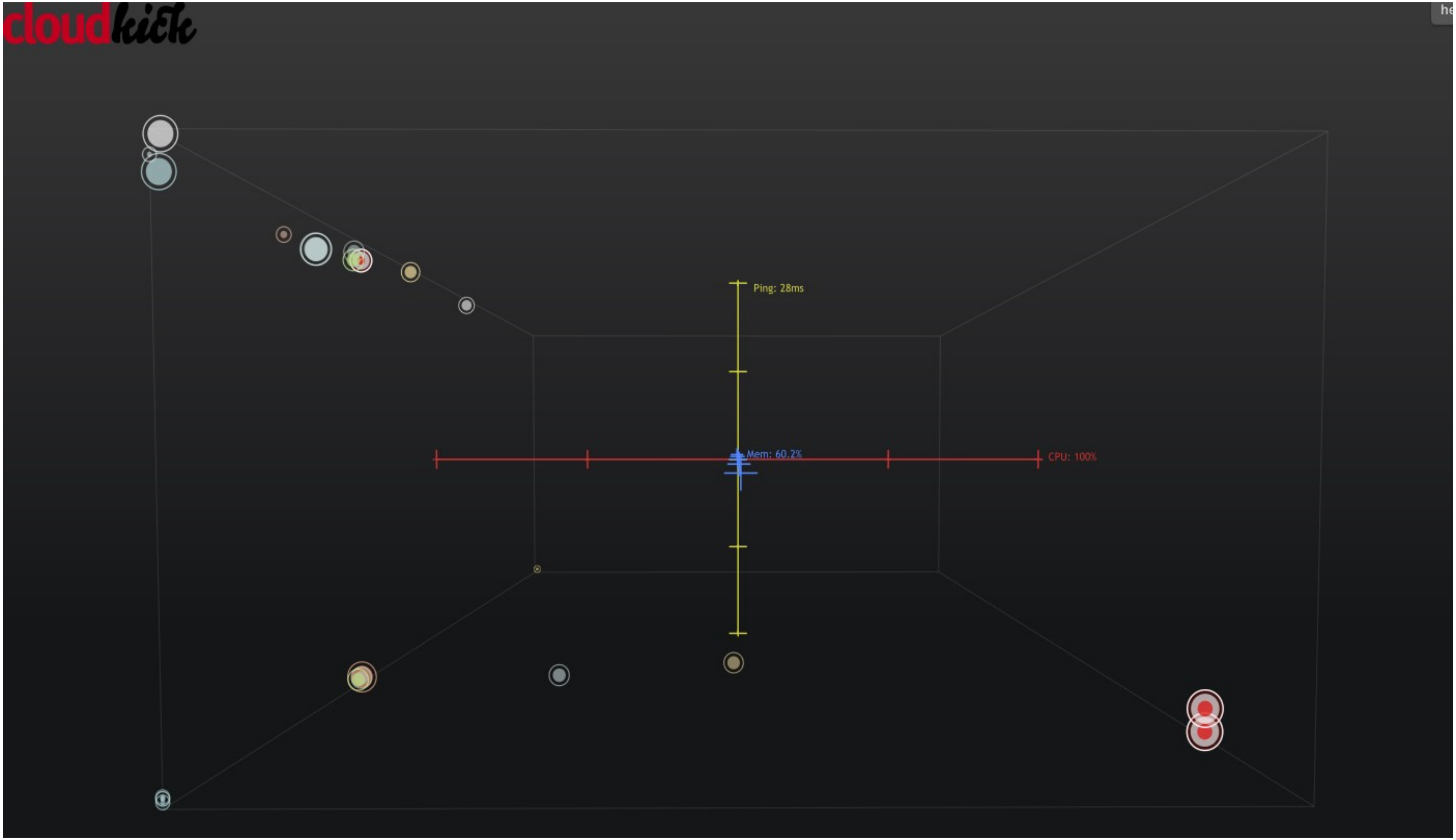
Noisy nodes

Node	Total alerts	Alerts/day

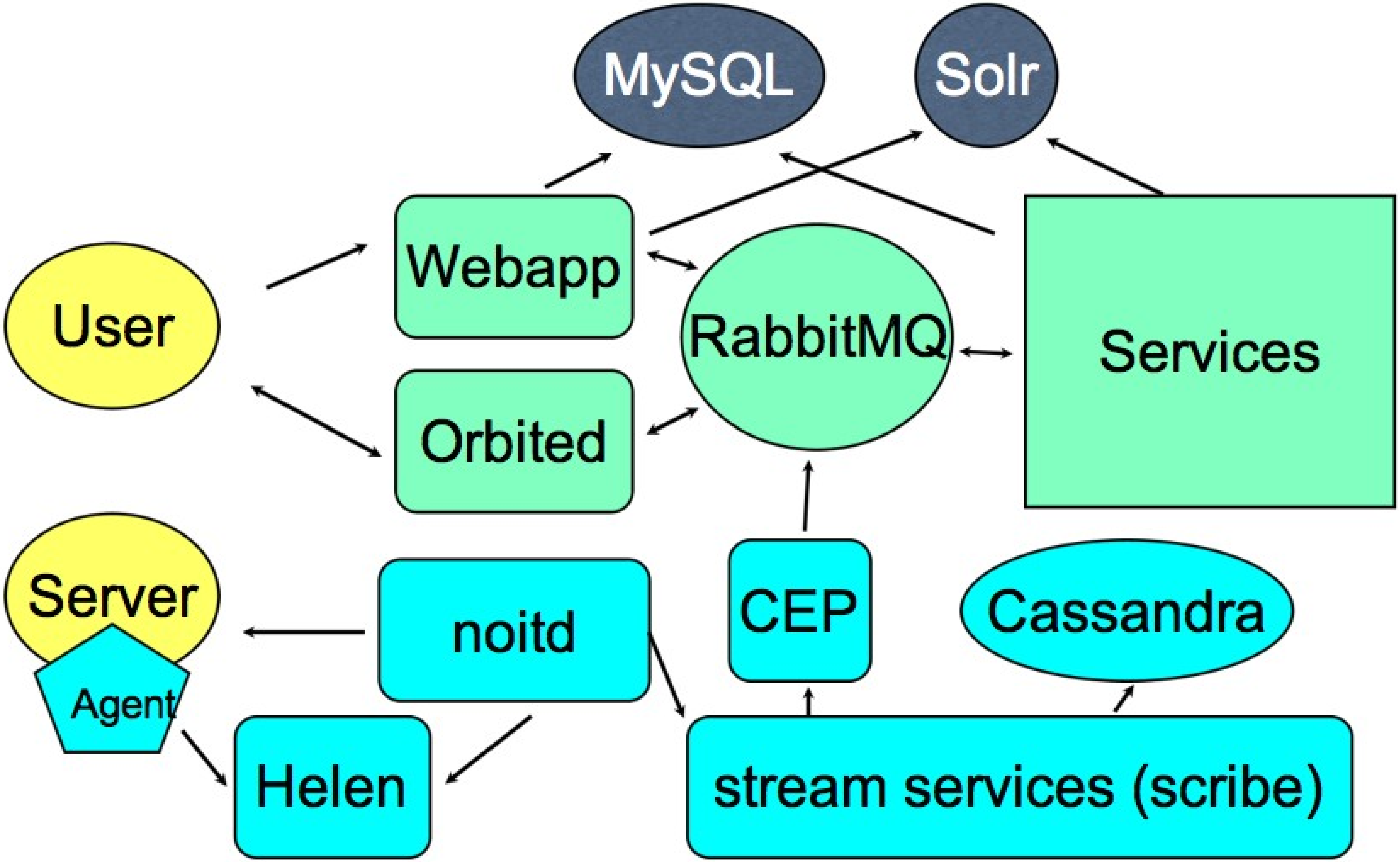
Quiet nodes

Node	Total alerts	Alerts/day

What is Cloudkick



Cloudkick Architecture



Why Python?

- Expressive
- Developers love it
- Quality library for just about anything you can think of
- Large and active community
- Whitespace matters!

Python at Cloudkick

- Used in many different places
- Web application
- Backend / Network services
- Random scripties

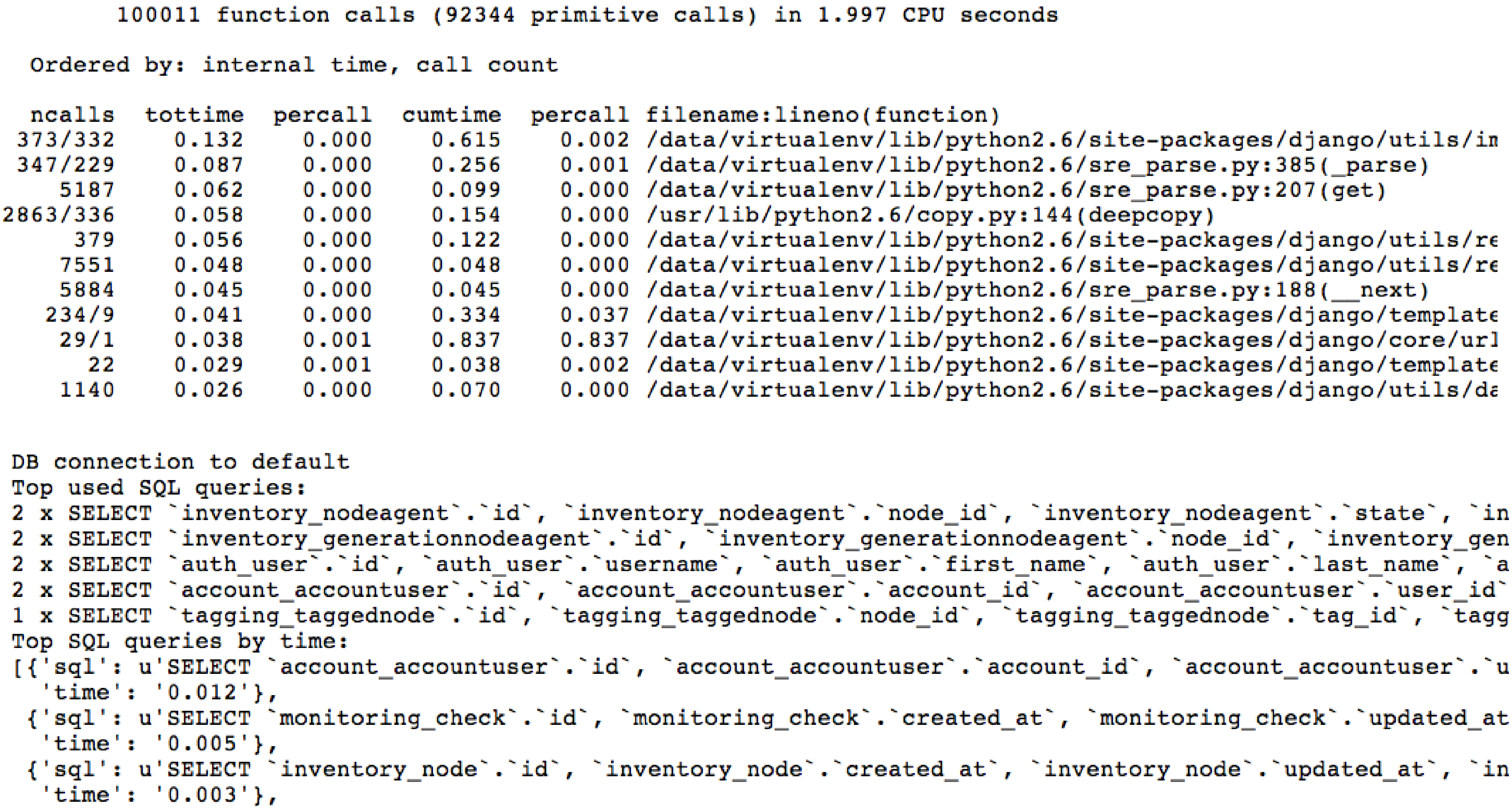
Web application & Django

- Web framework - Django
- Migrations - south
- API – piston
- Cassandra – Custom library & ORM
- Solr - solrpy

Web application & Django

- Profiling – ProfilingMiddleware, python-profiler
- Exception logging - django-sentry

Web application & Django



```
        100011 function calls (92344 primitive calls) in 1.997 CPU seconds

  Ordered by: internal time, call count

  ncalls  tottime  percall  cumtime  percall filename:lineno(function)
 373/332    0.132    0.000    0.615    0.002 /data/virtualenv/lib/python2.6/site-packages/django/utils/im
 347/229    0.087    0.000    0.256    0.001 /data/virtualenv/lib/python2.6/sre_parse.py:385(_parse)
    5187    0.062    0.000    0.099    0.000 /data/virtualenv/lib/python2.6/sre_parse.py:207(get)
2863/336    0.058    0.000    0.154    0.000 /usr/lib/python2.6/copy.py:144(deepcopy)
     379    0.056    0.000    0.122    0.000 /data/virtualenv/lib/python2.6/site-packages/django/utils/re
    7551    0.048    0.000    0.048    0.000 /data/virtualenv/lib/python2.6/site-packages/django/utils/re
    5884    0.045    0.000    0.045    0.000 /data/virtualenv/lib/python2.6/sre_parse.py:188(__next)
   234/9    0.041    0.000    0.334    0.037 /data/virtualenv/lib/python2.6/site-packages/django/template
    29/1    0.038    0.001    0.837    0.837 /data/virtualenv/lib/python2.6/site-packages/django/core/url
      22    0.029    0.001    0.038    0.002 /data/virtualenv/lib/python2.6/site-packages/django/template
    1140    0.026    0.000    0.070    0.000 /data/virtualenv/lib/python2.6/site-packages/django/utils/da


DB connection to default
Top used SQL queries:
2 x SELECT `inventory_nodeagent`.`id`, `inventory_nodeagent`.`node_id`, `inventory_nodeagent`.`state`, `in
2 x SELECT `inventory_generationnodeagent`.`id`, `inventory_generationnodeagent`.`node_id`, `inventory_gen
2 x SELECT `auth_user`.`id`, `auth_user`.`username`, `auth_user`.`first_name`, `auth_user`.`last_name`, `a
2 x SELECT `account_accountuser`.`id`, `account_accountuser`.`account_id`, `account_accountuser`.`user_id`
1 x SELECT `tagging_taggednode`.`id`, `tagging_taggednode`.`node_id`, `tagging_taggednode`.`tag_id`, `tagg
Top SQL queries by time:
[{'sql': u'SELECT `account_accountuser`.`id`, `account_accountuser`.`account_id`, `account_accountuser`.`u
  'time': '0.012'},
 {'sql': u'SELECT `monitoring_check`.`id`, `monitoring_check`.`created_at`, `monitoring_check`.`updated_at
  'time': '0.005'},
 {'sql': u'SELECT `inventory_node`.`id`, `inventory_node`.`created_at`, `inventory_node`.`updated_at`, `in
  'time': '0.003'},
```

Backend Services & Twisted

- Twisted is used heavily
- Old, battle-tested, (mostly) works
- Supports almost any protocol you can think of
- Writing good Twisted code is not that easy

Backend Services & Twisted

- Uncaught exceptions = memory leaks
- Be careful with threads.deferToThread
- emailLogObserver for sending tracebacks to email

Backend Services & Twisted

- Uncaught exceptions = memory leaks
- Be careful with threads.deferToThread
- emailLogObserver for sending tracebacks to email

Backend Services & Twisted

- We use a simple stats library for recording service-level metrics
- Metrics are exposed over HTTP in a JSON format
- Similar to https://github.com/codahale/metrics

Backend Services & Twisted

- counter.add(key)
- counter.inc_ops(key)
- counter.dec_ops(key)
- counter.add_avg(key, value)
- counter.bind(key, type, func, *args, **kwargs)
- @count_calls

Backend Services & Twisted

```
counter = Counter()

def my_function2(*args, **kwargs):
  try:
    # Increase operation count
    counter.inc_ops('execute_some_op')
    some_op()
  except Exception, e:
    # Increase the counter
    counter.add('some_op_failed')
  finally:
    # Signalize that the function has finished
    # pending -= 1
    counter.dec_ops('execute_come_op')
```

Backend Services & Twisted

```
counter = Counter()

def my_function3(*args, **kwargs):
  start = time.time()
  my_function()
  # Measure how long function execution took
  counter.add_avg('my_function_execute_ms',
                  (time.time() - start))
```

Backend Services & Twisted

```
{'metrics': [{'name': 'execute_some_op_pending',
              'type': 'int', 'value': 3},
             {'name': 'execute_some_op_total',
              'type': 'gauge', 'value': 4},
             {'name': 'my_function_execute_msg_avg',
              'type': 'float',
              'value': 22.666},
             {'name': 'my_function_execute_msg_max',
              'type': 'float',
              'value': 100},
             {'name': 'my_function_execute_msg_min',
              'type': 'float',
              'value': 4.2},
          {'name': 'some_op_failed', 'type': 'float', 'value': 3},
          {'name': 'uptime', 'type': 'float', 'value': 123.08233}],
 'state': 'ok',
 'status': 'service is good'}
```

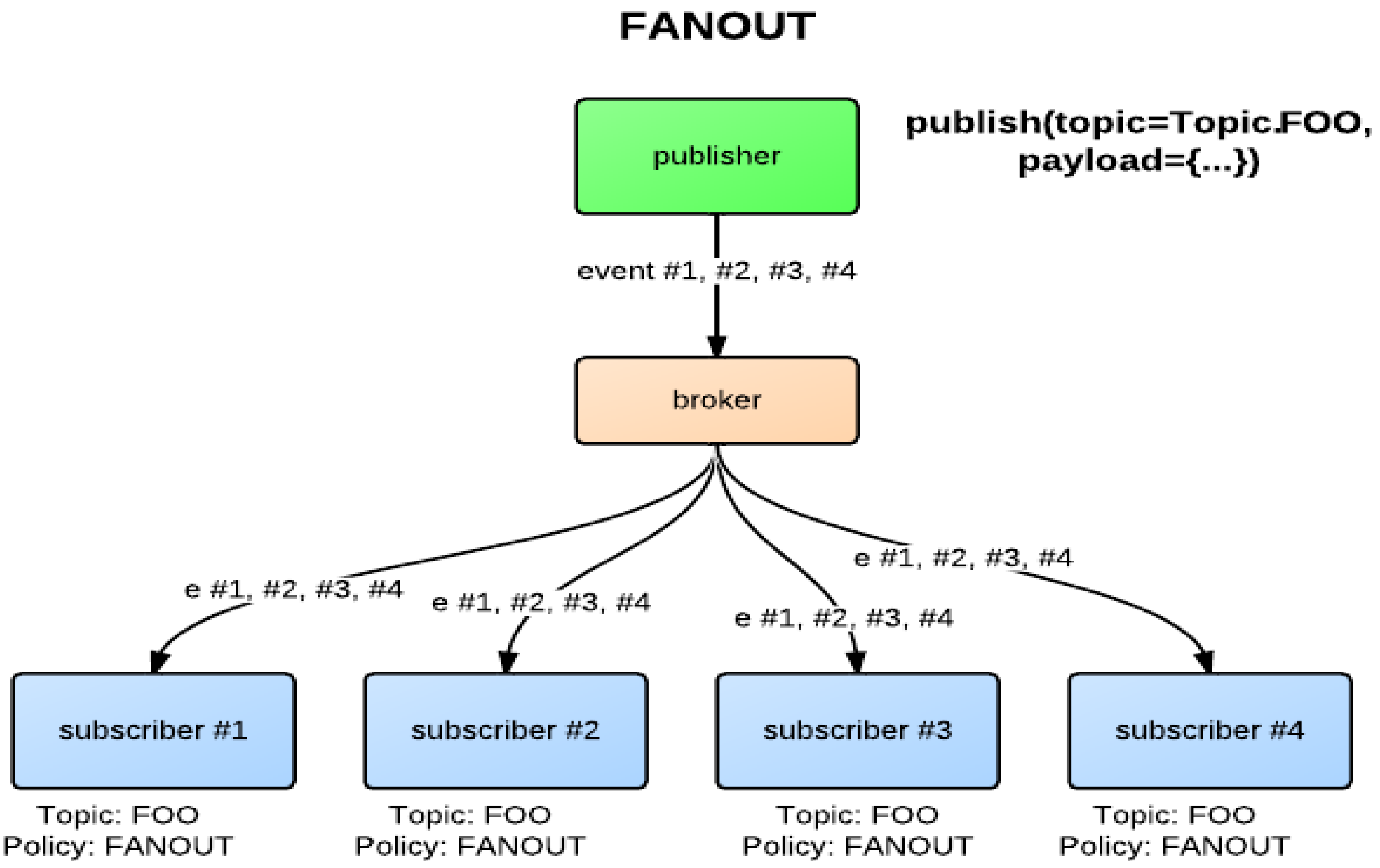
Service Communication and RPC

- Event framework
- Scribe
- Thrift

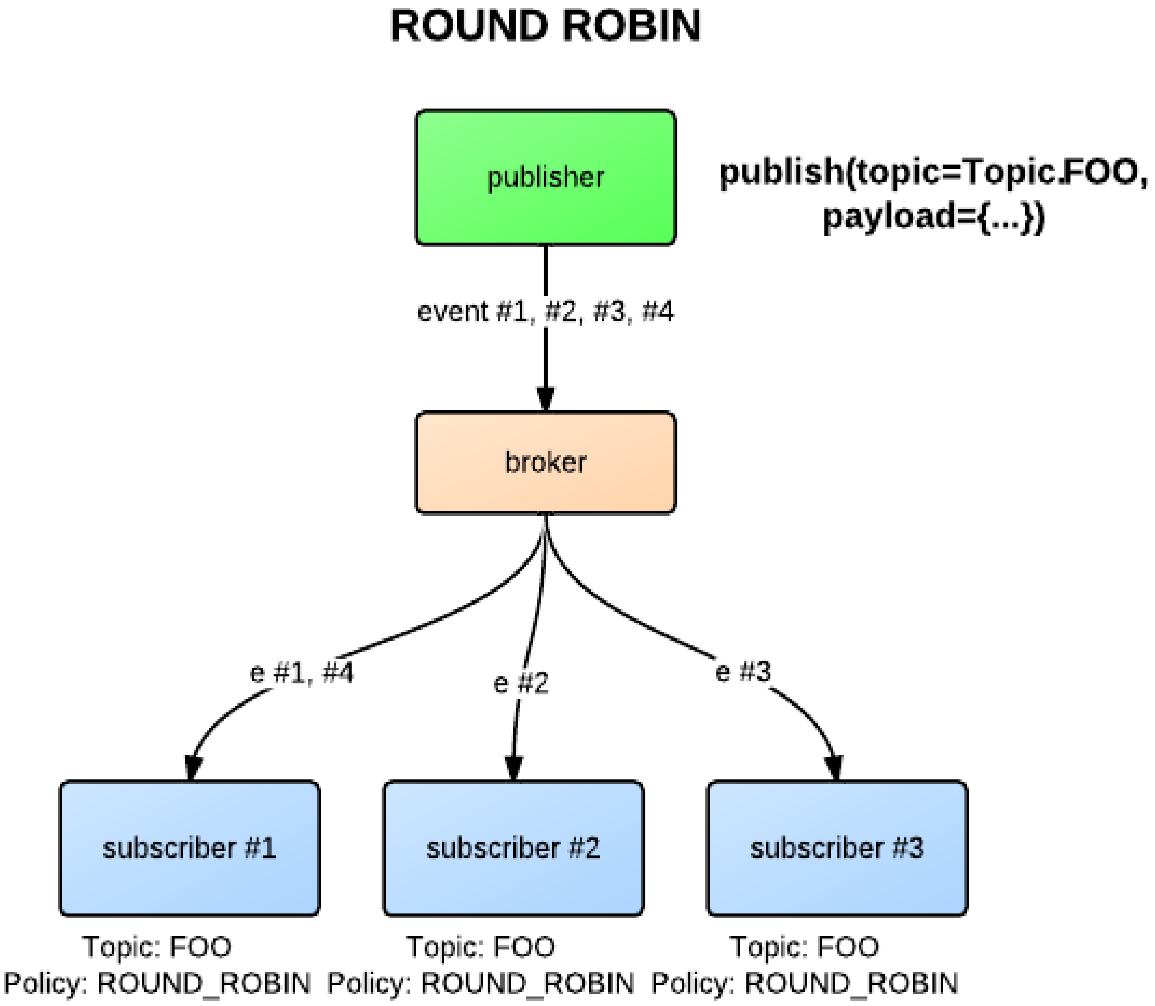
Service Communication and RPC

- Event framework
- Multiple policies:
  - Fan out
  - Round Robin
  - Scoped Round Robin
- Used in Django and Twisted land

Service Communication and RPC

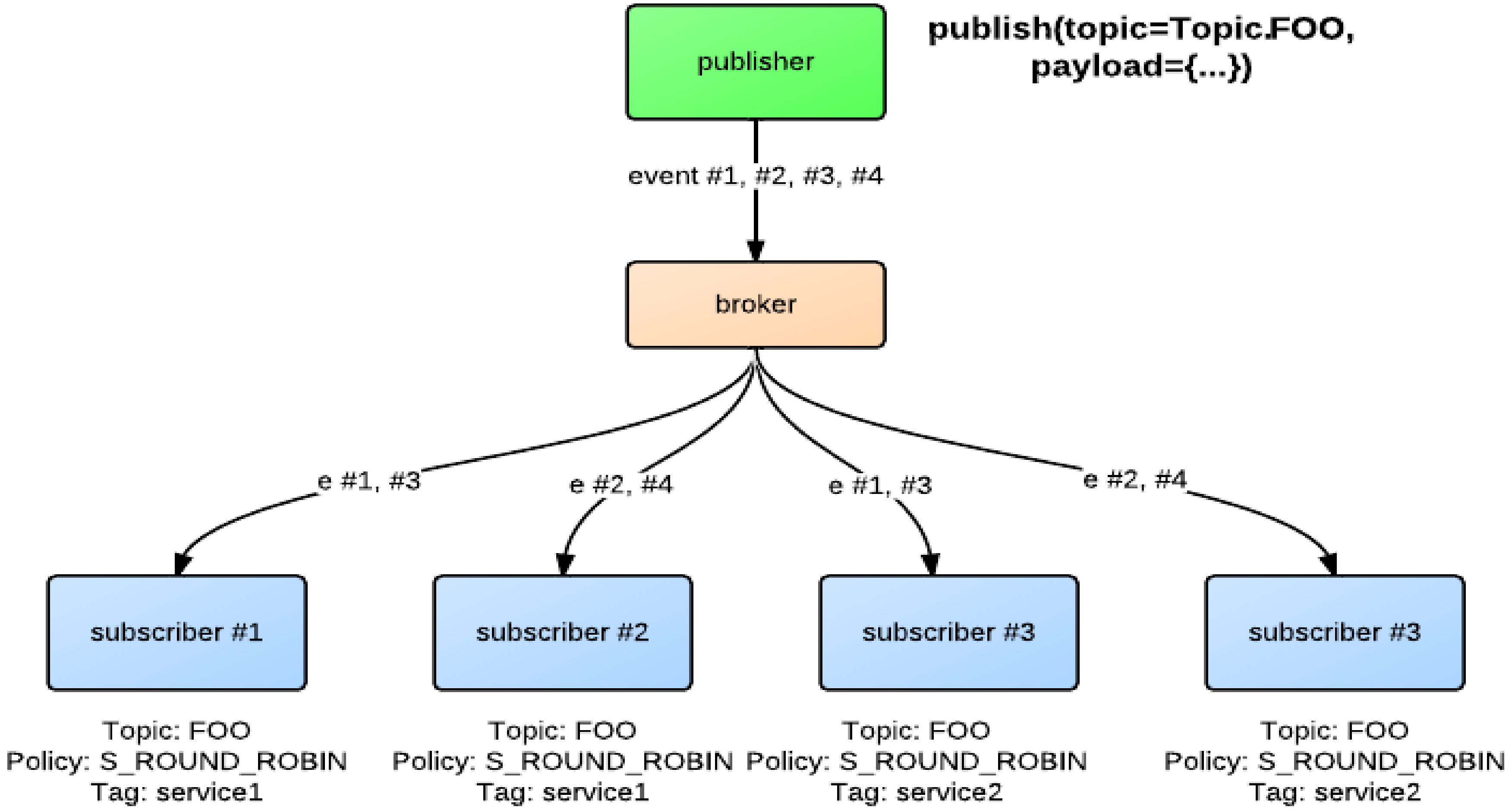


Service Communication and RPC



Service Communication and RPC

SCOPED ROUND ROBIN



publish(topic=Topic.FOO,
payload={...})

Topic: FOO
Policy: S_ROUND_ROBIN
Tag: service1

Topic: FOO
Policy: S_ROUND_ROBIN
Tag: service1

Topic: FOO
Policy: S_ROUND_ROBIN
Tag: service2

Topic: FOO
Policy: S_ROUND_ROBIN
Tag: service2

Service Communication and RPC

```
# Subscribe example

from cloudkick.events.async import dispatcher

def readyhook():
  dispatcher.register(Topic.TEST1, Policy.FANOUT,
                      handle_event1, topic)
  dispatcher.register(Topic.TEST2, Policy.ROUND_ROBIN,
                      handle_event2, topic)

dispatcher.ready_hook(readyhook)
```

Service Communication and RPC

```
# Publish example

from cloudkick.events.async import dispatcher
from cloudkick.events.path_defines import Topics

dispatcher.start()
event = event_ttypes.NodeNameUpdate(node=_id=1234,
                     old_name='foo', new_name='bar')
dispatcher.publish(Topics.NODE_UPDATE, event,
                   account_id=1, user_id=2)
```

Testing

- Not that fun
- Someone has to do it
- Functional tests – Django test framework & unittest
- Integration tests – Twisted test framework (trial)

Testing

- Custom Django and Twisted parallel test runner
- Up to 50% faster

Other places

- Deployment – fabric
- Continuous Integration – buildbot
- ...
- Snakes are everywhere!11

Questions?

?

Questions?

- Thanks

- P.S. We are looking for Python devs - http://rackertalent.com/san-francisco/