Monitoring an Openstack Deployment with Datadog

Who are we?

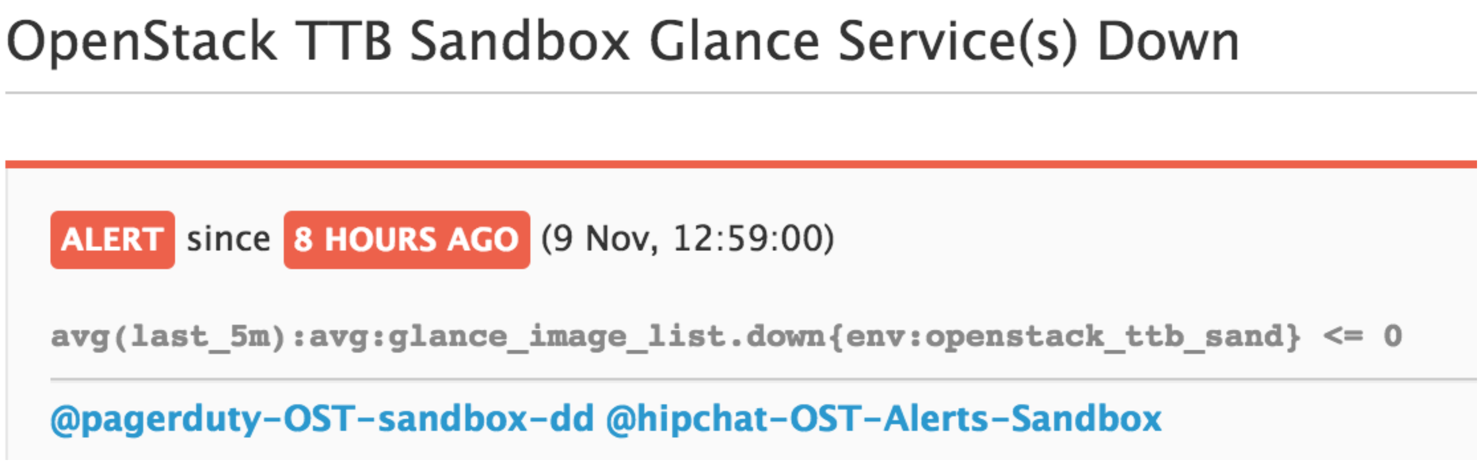
Target Enterprise Private Cloud Engineering Team

Matt Ahles	**Openstack Product Owner**
Aaron Chism	**Engineer**
Will Boege	**Architect**

What is Datadog?

Datadog is a SaaS Product used by the Target Openstack Team to provide insight into our systems and services

Performs monitoring and alerting of systems and services



OpenStack TTB Sandbox Glance Service(s) Down

ALERT since 8 HOURS AGO (9 Nov, 12:59:00)

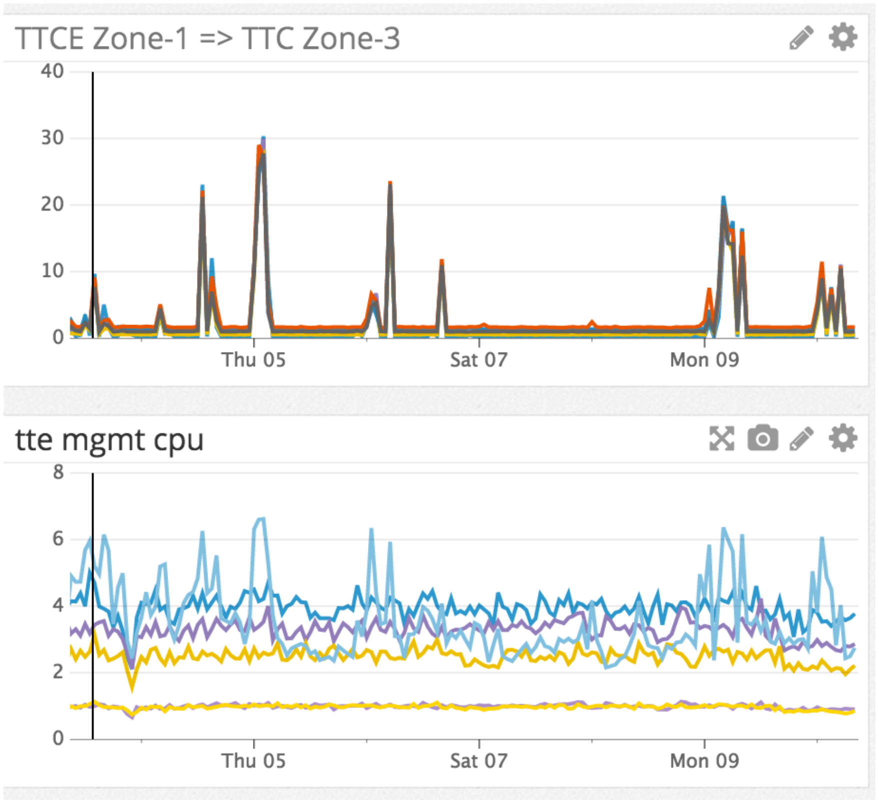
```
avg(last_5m):avg:glance_image_list.down{env:openstack_ttb_sand} <= 0
```

@pagerduty-OST-sandbox-dd @hipchat-OST-Alerts-Sandbox

What is Datadog?

Datadog is a SaaS Product used by the Target Openstack Team to provide insight into our systems and services

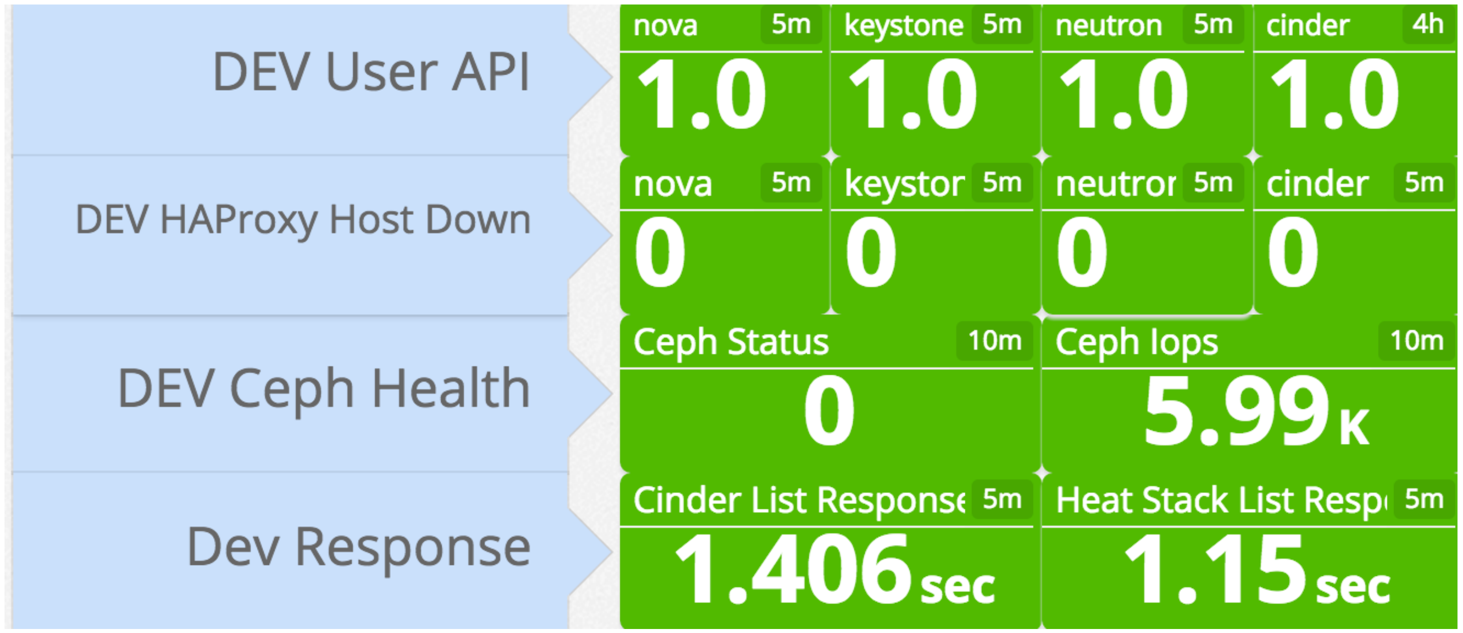
Analysis and correlation of system performance metrics



What is Datadog?

Datadog is a SaaS Product used by the Target Openstack Team to provide insight into our systems and services

Dashboarding

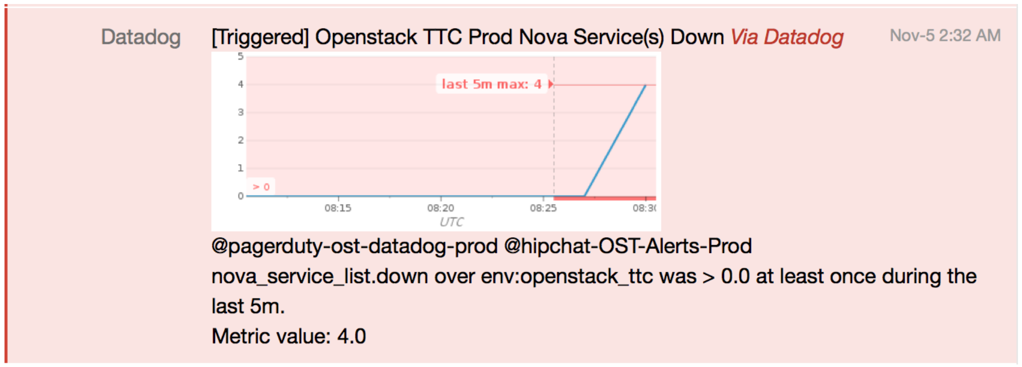


DEV User API	nova 5m: 1.0	keystone 5m: 1.0	neutron 5m: 1.0	cinder 4h: 1.0
DEV HAProxy Host Down	nova 5m: 0	keystor 5m: 0	neutror 5m: 0	cinder 5m: 0
DEV Ceph Health	Ceph Status 10m: 0		Ceph Iops 10m: 5.99K	
Dev Response	Cinder List Response 5m: 1.406sec		Heat Stack List Resp 5m: 1.15sec	

How Can Datadog Be Consumed?

The information held within data Datadog can be accessed via:

- WEB UI for direct consumption
- Restful API for querying data for use in other applications
- Integration modules that push data from Datadog into other tools to consume
    - Hipchat and Pagerduty is a good example of this.


Datadog [Triggered] Openstack TTC Prod Nova Service(s) Down *Via Datadog* Nov-5 2:32 AM

@pagerduty-ost-datadog-prod @hipchat-OST-Alerts-Prod
nova_service_list.down over env:openstack_ttc was > 0.0 at least once during the last 5m.
Metric value: 4.0

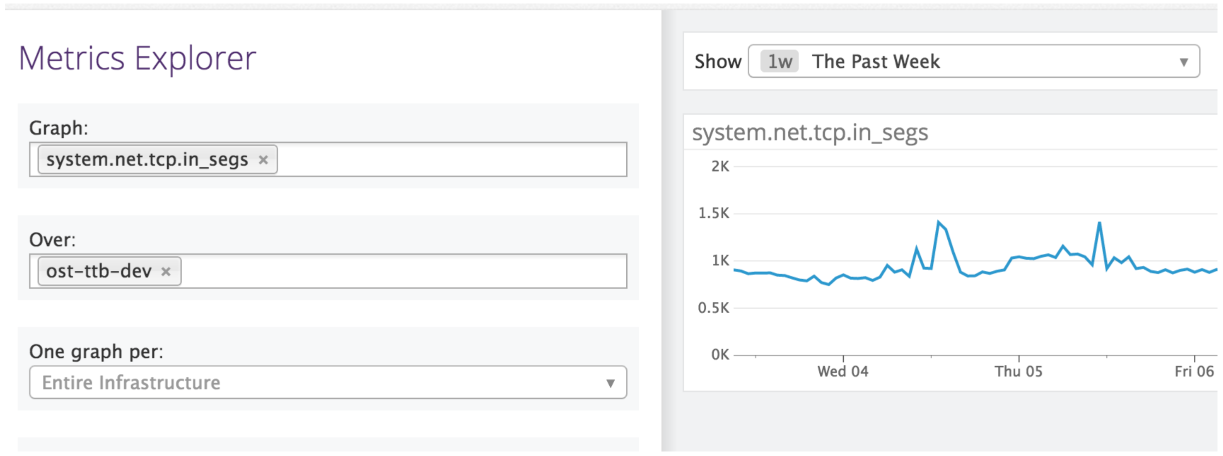
Data can be pushed into Datadog by several methods

- Traditional agent-based approach where a small application is installed on system
- Restful API to POST metrics
- Ruby/Python libraries that instantiate the API within a script or application
- Integration modules that interface with common infrastructure applications
    - Chef is a good example of this.

What We Like About Datadog

Like

- Very easy to get time series data from many disparate sources into Datadog to use for monitors/dashboards/charts.
  - I like to ‘screen scrape’ simple shell commands to push via the Ruby Gem
- Creating very functional dashboard layouts is simple
- *Most* of the integrations provided do an excellent job of extending the usefulness of the information contained within Datadog
  - *That being said – the native Openstack Integrations need a lot of work.*
- Monitoring and Alerting functions are fairly robust and integrate well into other applications.
- Robust ‘tagging’ system to group and aggregate metrics of like type.


Metrics Explorer

Graph: system.net.tcp.in_segs

Over: ost-ttb-dev

One graph per: Entire Infrastructure

Show: 1w The Past Week

What We Dislike About Datadog

Dislike

- Documentation is average at best
- Inability to use wildcards when selecting metrics to use in charting/alerting
  - Although, if you do your tagging right this isn't an issue
- Datadog only knows numbers, no ability to transform a number to a string in dashboarding. I.E. – cannot do something like (if val = 1 then (print "OK!")
  - *This can sometimes make dashboards fairly ambiguous to 'outsiders'*
- Simple host up/down monitors difficult to use due to false alarms from network disruptions/blips
- API key authorization model is somewhat 'all or nothing'

Datadog Demo

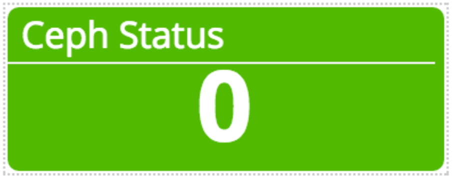
DEMO TIME!

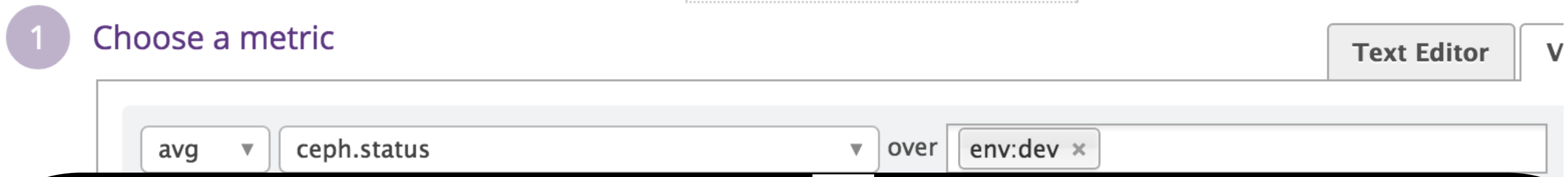
Ceph Monitors 101

Ceph has many options for administrators to obtain metrics from the cluster. With a few lines of Ruby, this output can be tracked and monitored within Datadog.

Here are a few simple examples that I have set up that have worked extremely well.

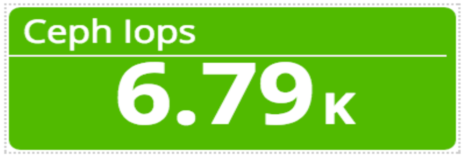
Ceph Monitors 101

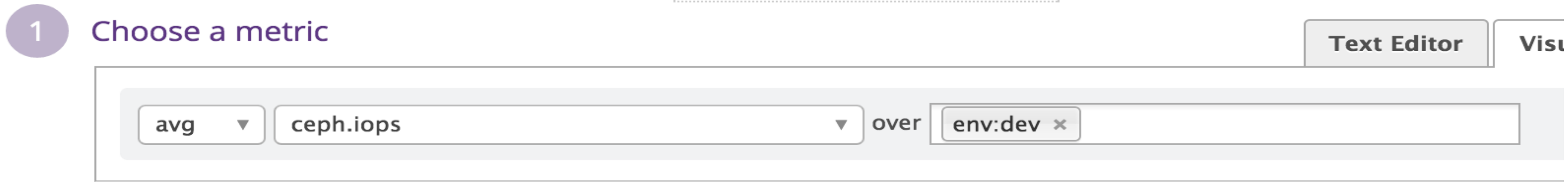




```
require 'rubygems'
require 'dogapi'

api_key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

health = `ceph health`
host = `hostname`

if host.include?("ttb")
      envname = "dev"
  elsif host.include?("ttc")
    envname = "prod-ttc"
  else
    envname = "prod-tte"
end
If health.include?("HEALTH_OK") then
  status = "0"
 elsif health.include?("WARN")
  status = "1"
 else
  status = "2"
end

dog = Dogapi::Client.new(api_key)
dog.emit_point("ceph.status", status[0], :tags => ["env:#{envname}","app:ceph"])
```

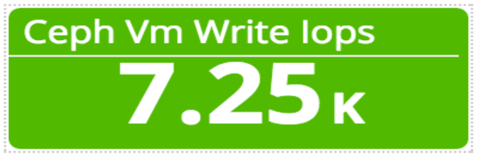
Ceph Monitors 101

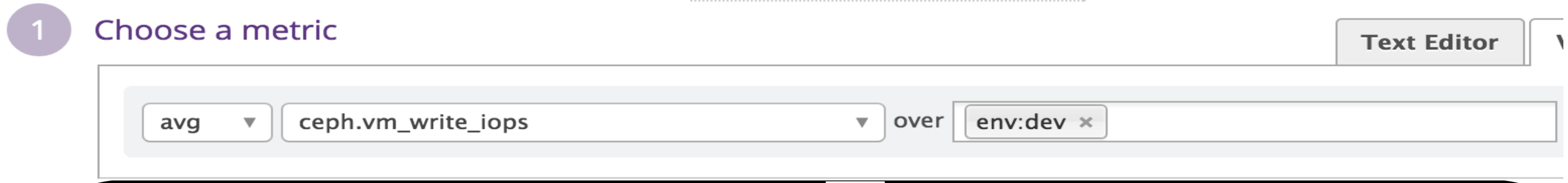




```
require 'rubygems'
require 'dogapi'

api_key = ”XXXXXXXXXXXXXXXXXXXXX"

iops = `ceph -s | grep client | awk '{print $9}'`
host = `hostname`

if host.include?("ttb")
     envname = "dev"
  elsif host.include?("ttc")
     envname = "prod-ttc"
  else
     envname = "prod-tte"
end

dog = Dogapi::Client.new(api_key)
dog.emit_point("ceph.iops", iops, :tags => ["env:#{envname}","app:ceph"])
```

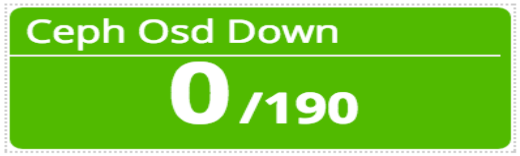
Ceph Monitors 101

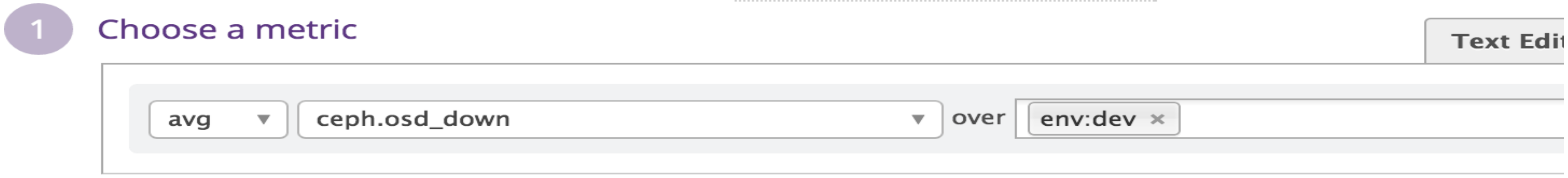




```
#!/bin/bash

# Generate Write Results
write_raw=$(fio --randrepeat=1 --ioengine=libaio --direct=1 --name=./test.write --filename=test \
--bs=4k --iodepth=4 --size=1G --readwrite=randwrite --minimal)

# Generate Read Results
read_raw=$(fio --randrepeat=1 --ioengine=libaio --direct=1 --name=./test.read --filename=test \
--bs=4k --iodepth=4 -size=1G --readwrite=randread --minimal)

writeresult_lat=$(echo $write_raw | awk -F\; '{print $81}')
writeresult_iops=$(echo $write_raw | awk -F\; '{print $49}')
readresult_lat=$(echo $read_raw | awk -F\; '{print $40}')
readresult_iops=$(echo $read_raw | awk -F\; '{print $8}')

ruby ./submit_lat_metrics.rb $writeresult_iops $readresult_iops $writeresult_lat $readresult_lat)
```

Ceph Monitors 101





```
require 'rubygems'
require 'dogapi'

api_key = “XXXXXXXXXXXXXXXXXXXXXXXXX"

dosd = `ceph osd tree | grep down | wc -l`
host = `hostname`

if host.include?("ttb")
        envname = "dev"
  elsif host.include?("ttc")
        envname = "prod-ttc"
  else
        envname = "prod-tte”
end

dog = Dogapi::Client.new(api_key)
dog.emit_point("ceph.osd_down", dosd, :tags => ["env:#{envname}","app:ceph"])
```