

Monitoring with Prometheus

SysAdmin ~3 years
WhitbySchool
448 Students,
K-8 IB & Montessori
Mac & Chromebook

whoami

Victor Vrantchan

twitter: **@wikiwalk**

<http://groob.io>



Robotics

& Programming

Monitoring Systems

Measure

Alert

Graph

Prometheus



http://prometheus.io



Prometheus

An open-source service monitoring system and time series database.

[Get Started](#)

Data model

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

[View details »](#)

Query language

A flexible query language allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

[View details »](#)

Visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, a GUI-based dashboard builder, and a console template language.

[View details »](#)

Storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

[View details »](#)

Operation

Each server is independent for reliability, relying only on local storage. Written in Go, all binaries are statically linked and easy to deploy.

[View details »](#)

Client libraries

Client libraries allow easy instrumentation of services. Currently, Go, Java, and Ruby are supported. Custom libraries are easy to implement.

[View details »](#)

Alerting

Alerts are defined based on Prometheus's flexible query language and maintain dimensional information. An alertmanager handles notifications and silencing.

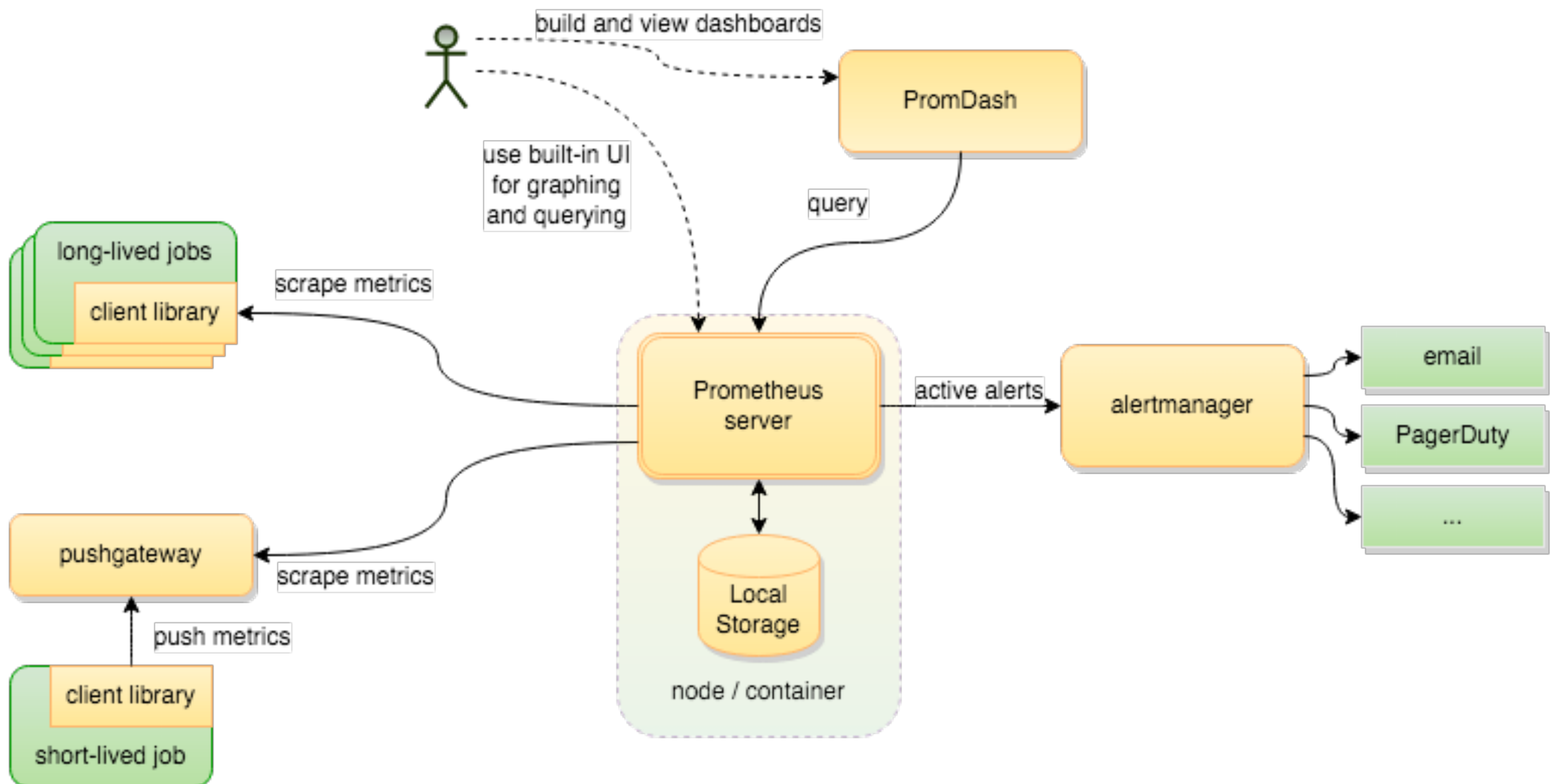
Exporters

Existing exporters allow bridging of third-party data into Prometheus. Examples: system statistics, as well as Docker, HAProxy, StatsD, and JMX metrics.

[View details »](#)

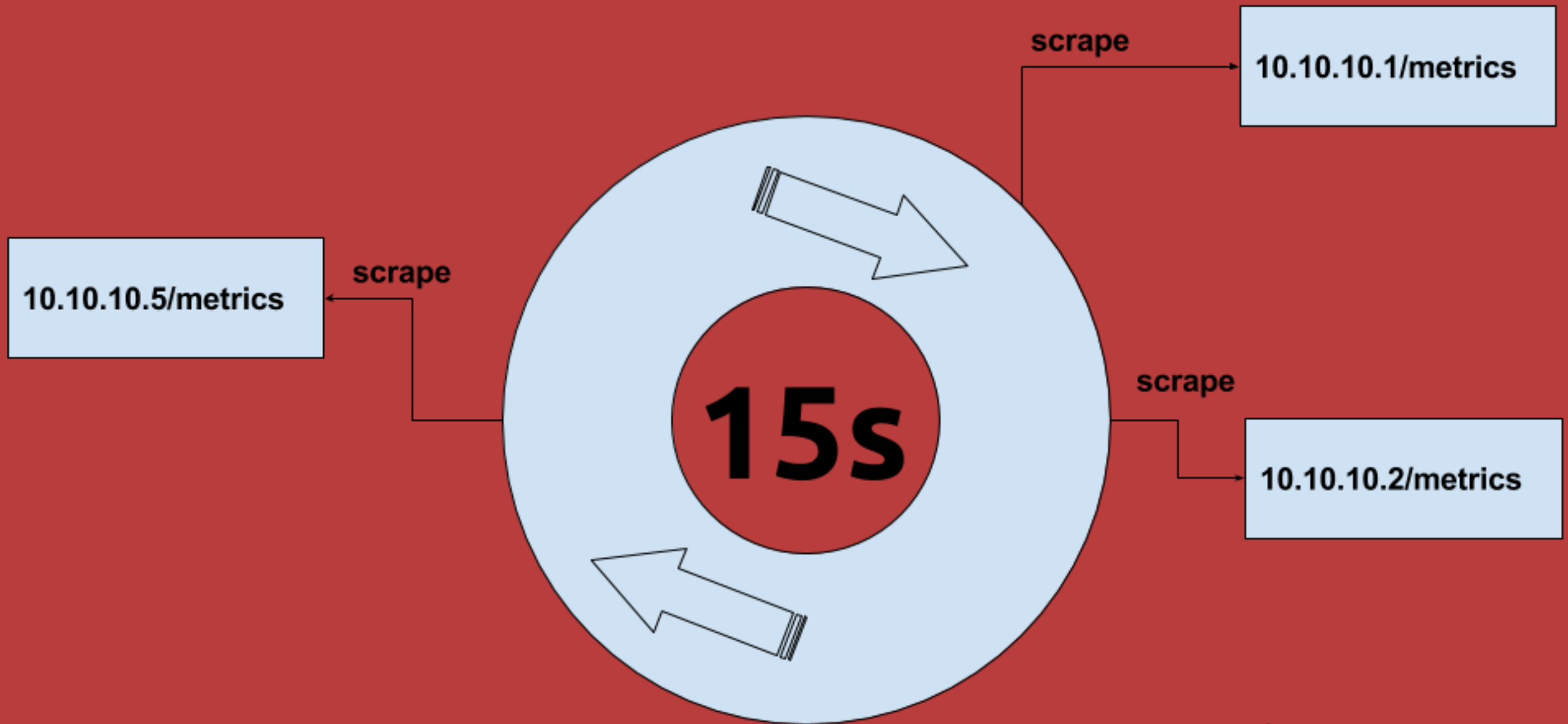
Time Series Database & Monitoring System

Architecture



Measure

/metrics



/metrics

HELP container_cpu_usage_seconds_total Cumulative cpu time consumed per cpu in seconds.

TYPE container_cpu_usage_seconds_total counter

container_cpu_usage_seconds_total{cpu="cpu00",id="/docker/7a2bff1",name="prometheus"} 0.344511525

/metrics

HELP container_cpu_usage_seconds_total Cumulative cpu time consumed per cpu in seconds.

TYPE container_cpu_usage_seconds_total counter

container_cpu_usage_seconds_total{cpu="cpu0",id="/docker/7a2bfff1",name="prometheus"} 0.344511525

<metric name>{<label name>=<label value>, ...}

/metrics

HELP container_cpu_usage_seconds_total Cumulative cpu time consumed per cpu in seconds.

TYPE container_cpu_usage_seconds_total counter

container_cpu_usage_seconds_total{cpu="cpu00",id="/
docker/7a2bfff1",name="prometheus"}
0.344511525

<metric name>{**<label name>=<label
value>, ...}**

/metrics

HELP container_cpu_usage_seconds_total Cumulative cpu time consumed per cpu in seconds.

TYPE container_cpu_usage_seconds_total counter

container_cpu_usage_seconds_total{cpu="cpu00",id="/docker/7a2bff1",name="prometheus"} **0.344511525**

/metrics

Plaintext over HTTP

Client Libraries

Go

Java or Scala

Python

Ruby

Bash

Haskell

Node.js

.NET / C#

client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()
```

```
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)
```

```
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)
```

```
timestamp.set_to_current_time()
```

```
installs.labels("my_mac").set(5)
```

```
installs.labels("my_mac").inc()
```

```
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a  
batch job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


client_python

```
from prometheus_client import CollectorRegistry, Gauge,  
push_to_gateway
```

```
registry = CollectorRegistry()  
timestamp = Gauge('job_last_success_unixtime', 'Last time a batch  
job successfully finished', registry=registry)  
installs = Gauge('number_of_installs', 'number of munki pkgs  
installed', ["node"], registry=registry)  
timestamp.set_to_current_time()  
installs.labels("my_mac").set(5)  
installs.labels("my_mac").inc()  
push_to_gateway('pushgateway.cluster.local:9091', job='munki',  
registry=registry)
```


Push Gateway

Pushgateway

Metrics

Status

▼ job="munki"

Delete Group

▼ job_last_success_unixtime **Last time a batch job successfully finished** GAUGE last pushed: 2015-11-06 13:49:17.519950162 +0000 UTC

▼ number_of_installs **number of munki pkgs installed** GAUGE last pushed: 2015-11-06 13:49:17.519950162 +0000 UTC

Labels	Value	Timestamp
instance="" job="munki" node="my_mac"	6	

Exporters

Node Exporter
AWS CloudWatch exporter
Blackbox exporter
Collectd exporter
Consul exporter
Graphite exporter
HAProxy exporter
JMX exporter
Mesos task exporter
MySQL server exporter
SNMP exporter
StatsD exporter

Apache exporter
BIND exporter
CouchDB exporter
Django exporter
mtail
Heka exporter
HTTP(s)/TCP/ICMP
blackbox prober
Memcached exporter
Meteor JS web framework
exporter
Minecraft exporter module
MongoDB exporter
Munin exporter
New Relic exporter
PostgreSQL exporter

RabbitMQ exporter
Redis exporter
RethinkDB exporter
Rsyslog exporter
collector exporter
SMTP/Maildir MDA blackbox
prober
SQL query result set metrics
exporter

mtail

```
# simple line counter  
counter line_count  
/$/ {  
    line_count++  
}
```


mtail

2. zsh

```
counter caching_parsed_log_lines
counter caching_registrations
counter caching_cleanups
counter caching_requests by request_source, file_type

/^(?P<date>\d+-\d+-\d+ \d+:\d+:\d+\.\d+)/{
    strftime($date, "2006-01-02 15:04:05.000")
    caching_parsed_log_lines++

    # Registration
    /(\bRegistration\b \bsucceeded\b\.) / {
        caching_registrations++
    }
    # Cleanup
    /(\bCleanup\b \bsucceeded\b\.) / {
        caching_cleanups++
    }

    # Requests
    /\#.* Received GET request from .+? \[(?P<request_source>\w+\/\d+\.\d+).+? for .+\. (?P<file_type>\w+)/ {
        caching_requests[$request_source][$file_type]++
    }
}
```

~ >>>

REGULAR EXPRESSION

1 MATCH - 376 STEPS

/ \#.* Received GET request from .+? \[(?P<request_source>\w+\/\d+\.\d+)\].+? for .+\.(?P<file_type>\w+)

gmixXsuUAJ ?

TEST STRING

#1JTjbr741a4f Received GET request from 127.0.0.1:54067 [MacAppStore/2.1 (Macintosh; OS X 10.11.1; 15B42) AppleWebKit/1601.2.7.2] for /apple-assets-us-std-000001/Purple69/v4/0d/42/06/0d420659-b758-b234-56cf-ddb71225b2cd/rrl6911432689796070578.pkg

MATCH INFORMATION

MATCH 1

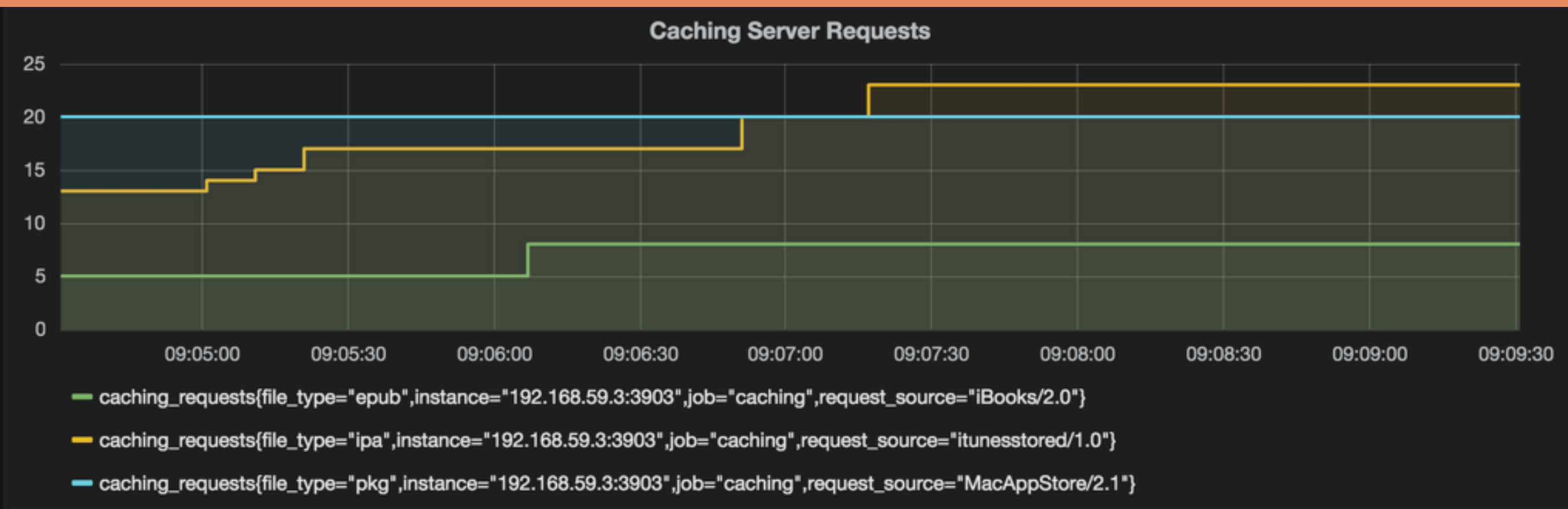
request_source [57-72] `MacAppStore/2.1`

file_type [243-246] `pkg`

Caching Server Exporter

```
# HELP caching_data data cached by server.  
# TYPE caching_data gauge  
caching_data{type="Books"} 3.9385975e+07  
caching_data{type="Mac Software"} 7.3859516e+07  
caching_data{type="Movies"} 0  
# HELP caching_status_active caching server is currently  
running  
# TYPE caching_status_active gauge  
caching_status_active 1
```


Caching Server Exporter





Investigate

credit:
@david__jones

Query Language

```
topk(3,  
sum(rate(container_cpu_user_seconds_total[5m]  
))  
by (name,instance))
```


Expression (press Shift+Enter for newlines)

Execute - insert metric at cursor -

Graph Console

Element	Value
no data	

Add Graph

Expression (press Shift+Enter for newlines)

Execute

- insert metric at cursor -

Alerts

ALERT DNSRequestRate

IF rate(skydns_dns_request_count[10s]) > 10

FOR 30s

**SUMMARY "High DNS Request rate on
{{\$labels.instance}}"**

**DESCRIPTION "The DNS server
{{\$labels.instance}} request rate is above (current
value: {{\$value}})"**

ALERT DhcpScopeAlmostFull

IF ceil((dhcp_leases_current_count / dhcp_leases_max_count)*100) > 90 FOR 2m

SUMMARY "DHCP scope {{\$labels.network}} is almost full"

DESCRIPTION "DHCP scope {{\$labels.network}} is {{\$value}}% full"

Prometheus



Alertmanager



pagerduty




email





Slack



```
notification_config {  
  name: "alertmanager_slack"  
  slack_config {  
    webhook_url: "https://hooks.slack.com/services/  
000000/BOJDXCFV/0000012234"  
    channel: "mactechconf"  
  }  
}
```

```
aggregation_rule {  
  repeat_rate_seconds: 300  
  notification_config_name: "alertmanager_slack"  
}
```



%1


%2


%3


%4

WhitbySchool
● Victor Vrantchan

★ STARRED

caedmon

cais-comms-cte

communications


general


it-public


le-team


...

 **prometheus-test** ▾

 1









Today

firing

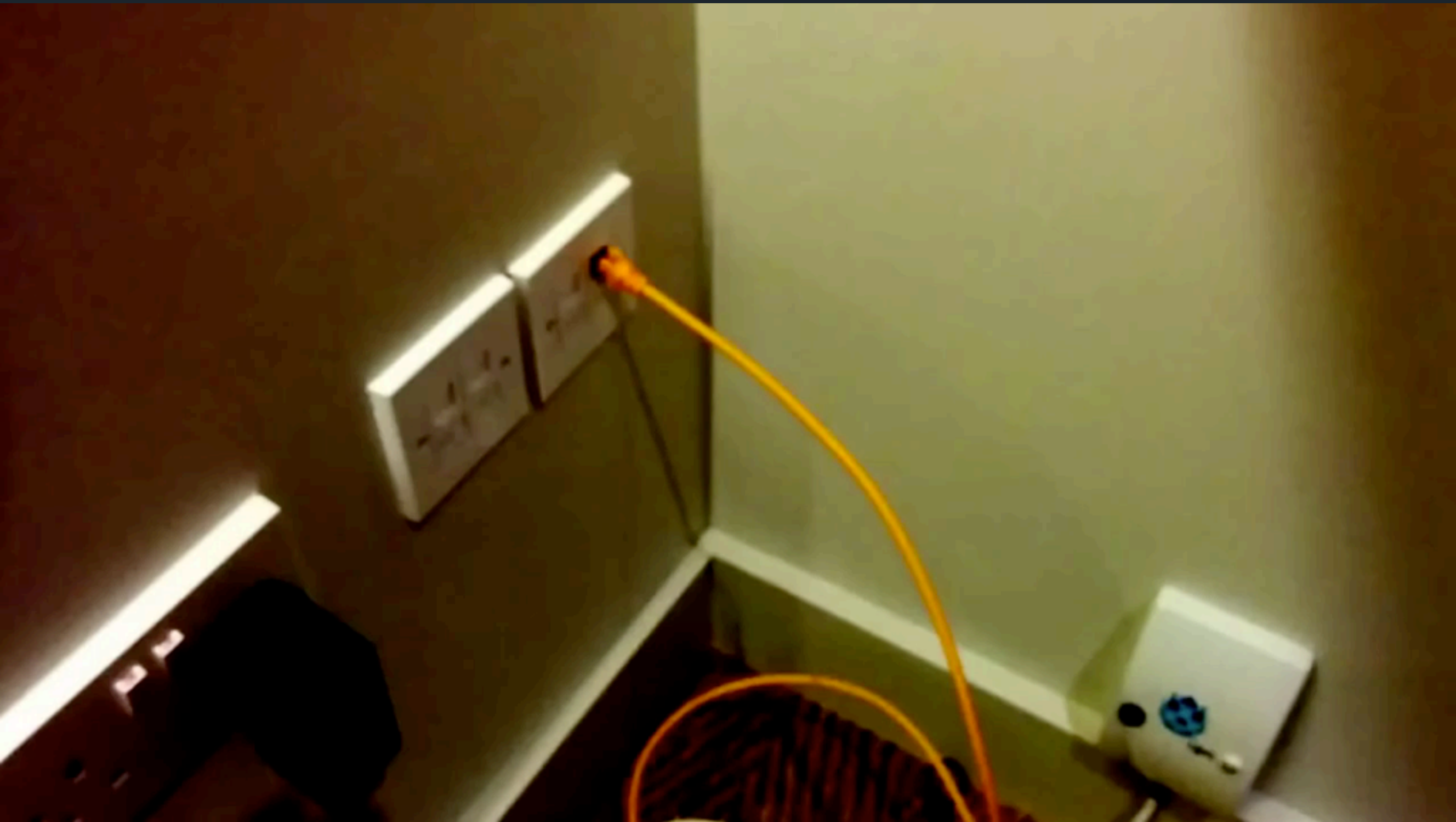
InternetDown

firewall down

firewall internet connection down

Status

firing



Demo by Brian Brazil

“Every time my pager goes off, I should be able to **react with a sense of urgency**. I can only do this a few times a day before I get fatigued.”

- *Rob Ewaschuk*



Graphs






 Home ▾



Zoom Out

 Last 6 hours



Dashboards



Data Sources



Starred dashboards

Dashboards



admin



Main Org. ▾



Grafana admin

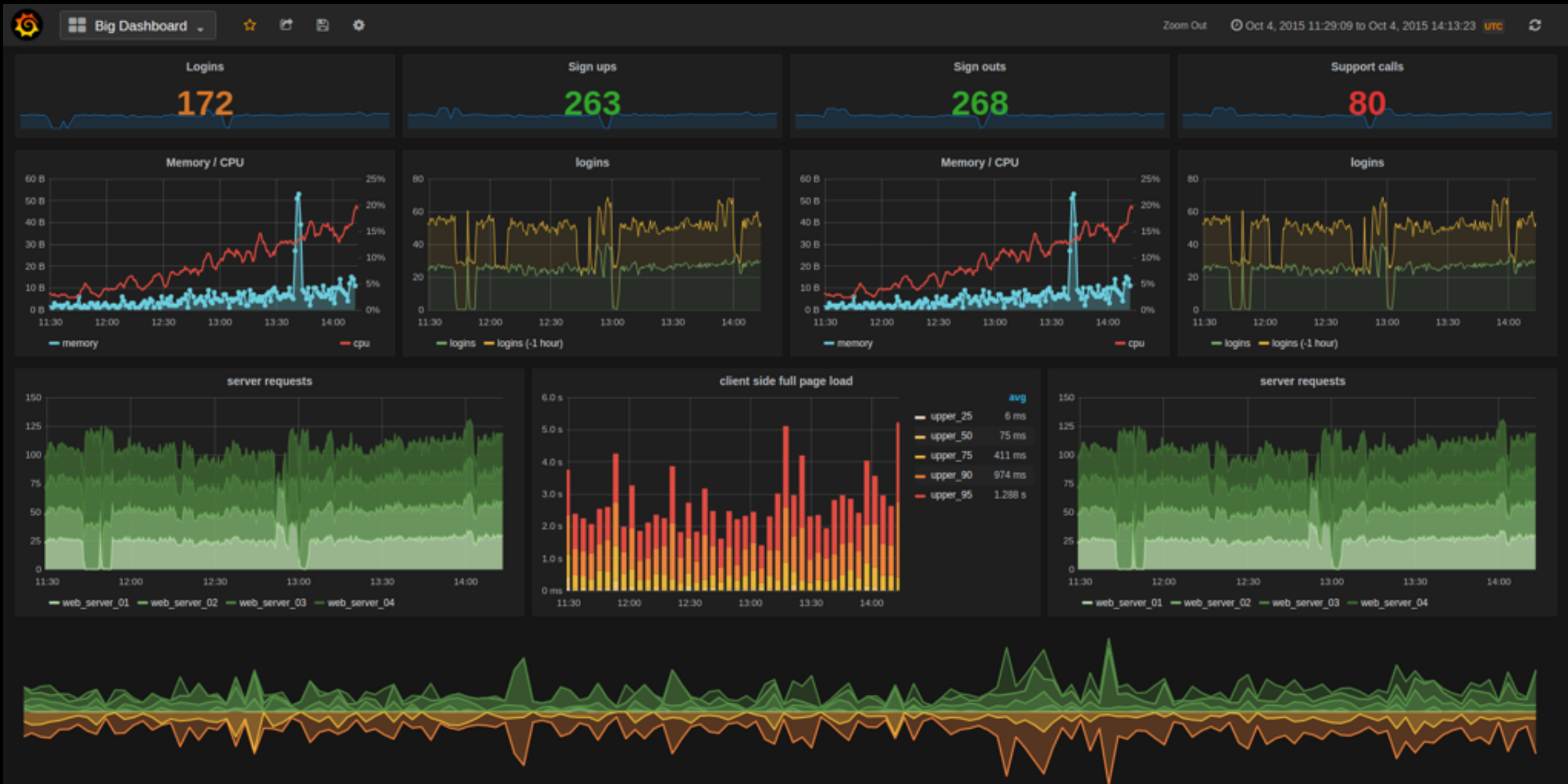


Sign out

CHARTS!!!!!!!



CHARTS EVERYWHERE



[DEMO]

#monitoring

References

<http://groob.io/posts/>

https://github.com/groob/caching_exporter

<http://www.robustperception.io/blog/>

<https://puppetlabs.com/presentations/practical-guide-monitoring-and-alerting-timeseries-scale>

<http://smcdermott.tumblr.com/post/48823881206/rob-ewaschuk-my-philosophy-on-alerting>

<http://prometheus.io/blog/2015/06/24/monitoring-dreamhack/>

<http://www.boxever.com/tag/monitoring>