



# Monitoring MySQL with Prometheus & Grafana

Julien Pivotto (@roidelapluie)

Percona University Belgium

June 22nd, 2017

# SELECT USER();

- Julien "roidelapluie" Pivotto
- @roidelapluie
- Sysadmin at [inuits](#)
- Automation, monitoring, HA
- MySQL/MariaDB user/admin/contributor
- Grafana and Prometheus user/contributor





inuits

A photograph showing a dense cluster of many traffic lights, all mounted on a single pole. The lights are oriented in various directions, some showing red, others green or yellow. They are set against a clear blue sky. A dark rectangular box is overlaid on the bottom center of the image.

# Monitoring

# DevOps

## 4 keys principles

- Culture
- Automation
- **Metrics**
- **Sharing**

*Damon Edwards and John Willis, 2010*



# Metrics matters

- Find when a service is unavailable (*before your users*)
- Understand Failures (*post-mortem*)
- Learn from your infrastructure
- Anticipate



# Collecting Metrics

- In general, you should collect as many metrics as you can
- Frequently (5 min collection are not enough)
- Store the metrics in a sane way (avoid averaging old metrics, etc)



# How to collect metrics?

- Many many solutions, including:
  - Graphite
  - Elasticsearch beats
  - Zabbix
  - and .. Prometheus



# Out of scope today

- Alerting
- Prometheus and Grafana installation
- Prometheus Fine Tuning

But I'm happy to discuss that after the talks!





# Prometheus

<https://prometheus.io/>

# Prometheus

Prometheus is a *Cloud-Native Data-Centric Open-Source Performant Simple metrics collection, analysis and alerting tool.*

Nothing more.



# Cloud Native

- Easy to configure, deploy, maintain
- Designed in multiple services
- Container ready
- Orchestration ready (dynamic config)
- Fuzziness



# Data Centric

A Metric in Prometheus has metadata:

```
mysql_global_status_handlers_total{handler="tmp_write"} 1122
```

And lots of function to filter, change, remove...  
those metadata while fetching them.



# Open Source

- Apache 2.0
- Go
- Support for multiple OS
- Many "exporters":

<https://github.com/prometheus/prometheus/wiki/Default-port-allocations>



# Performance

- Prometheus is designed to fetch data in an interval measured in SECONDS
- You can fine tune its memory usage and when it flushes to disk
- It can also adapt its scraping frequency dynamically



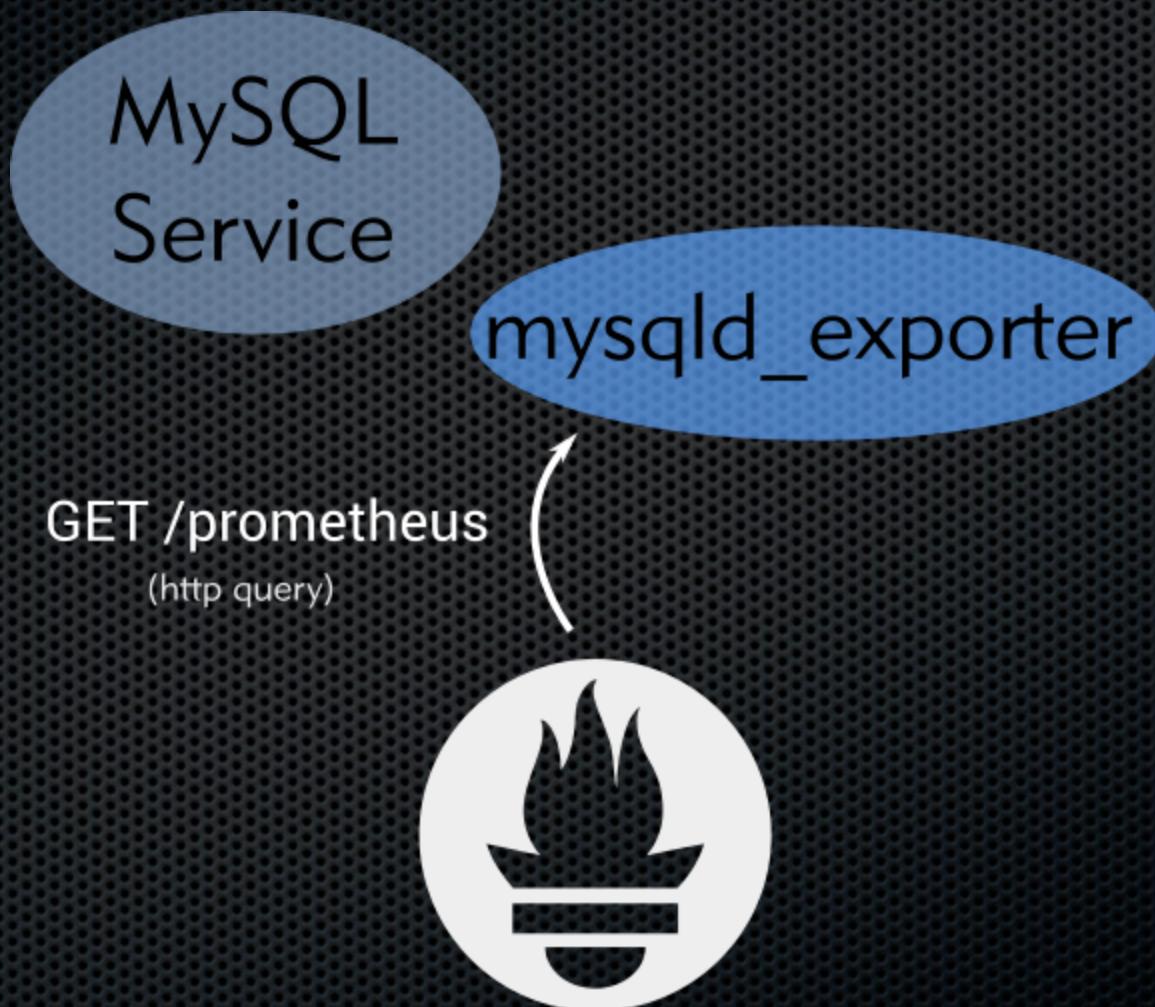
# How does it work?

MySQL  
Service

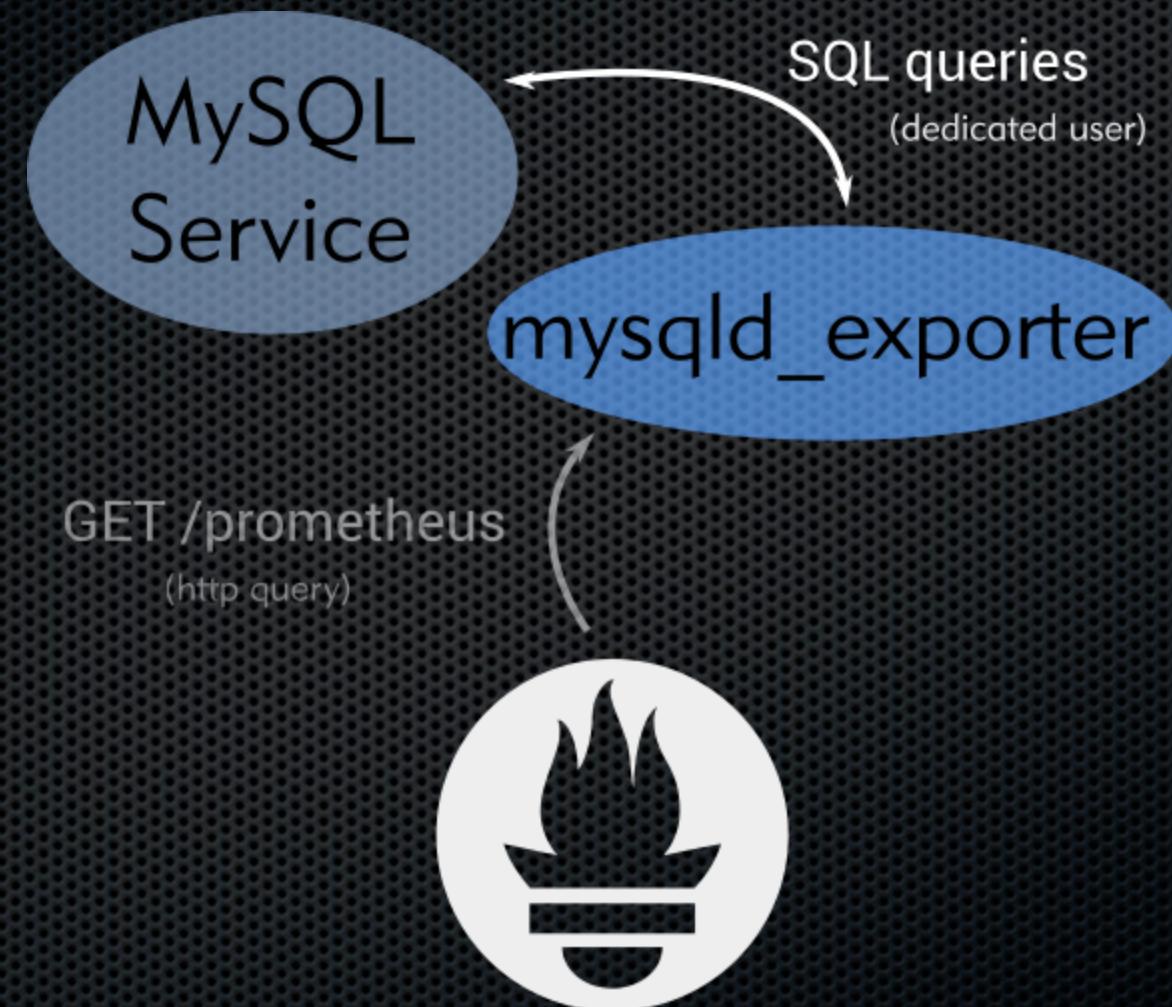
mysqld\_exporter



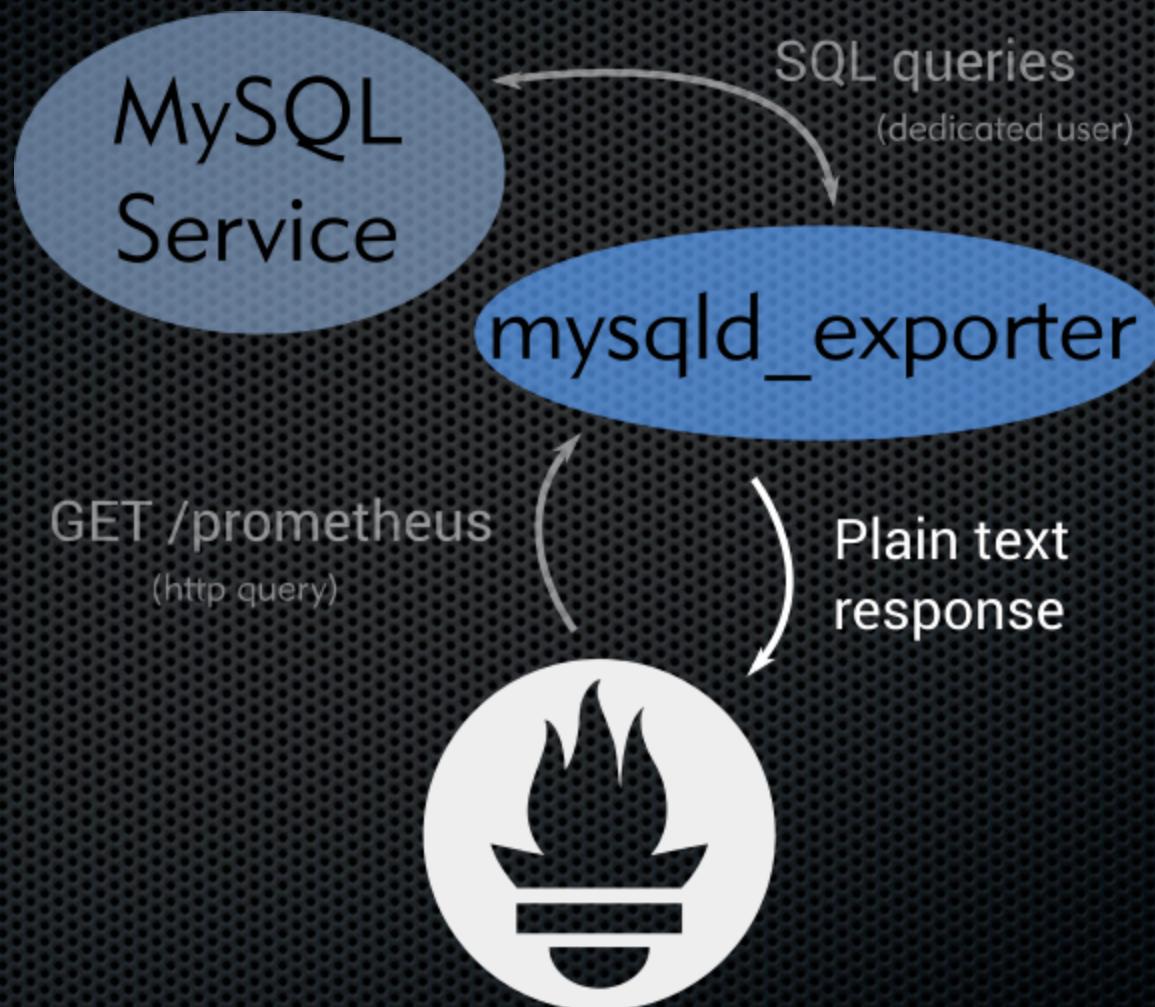
# How does it work?



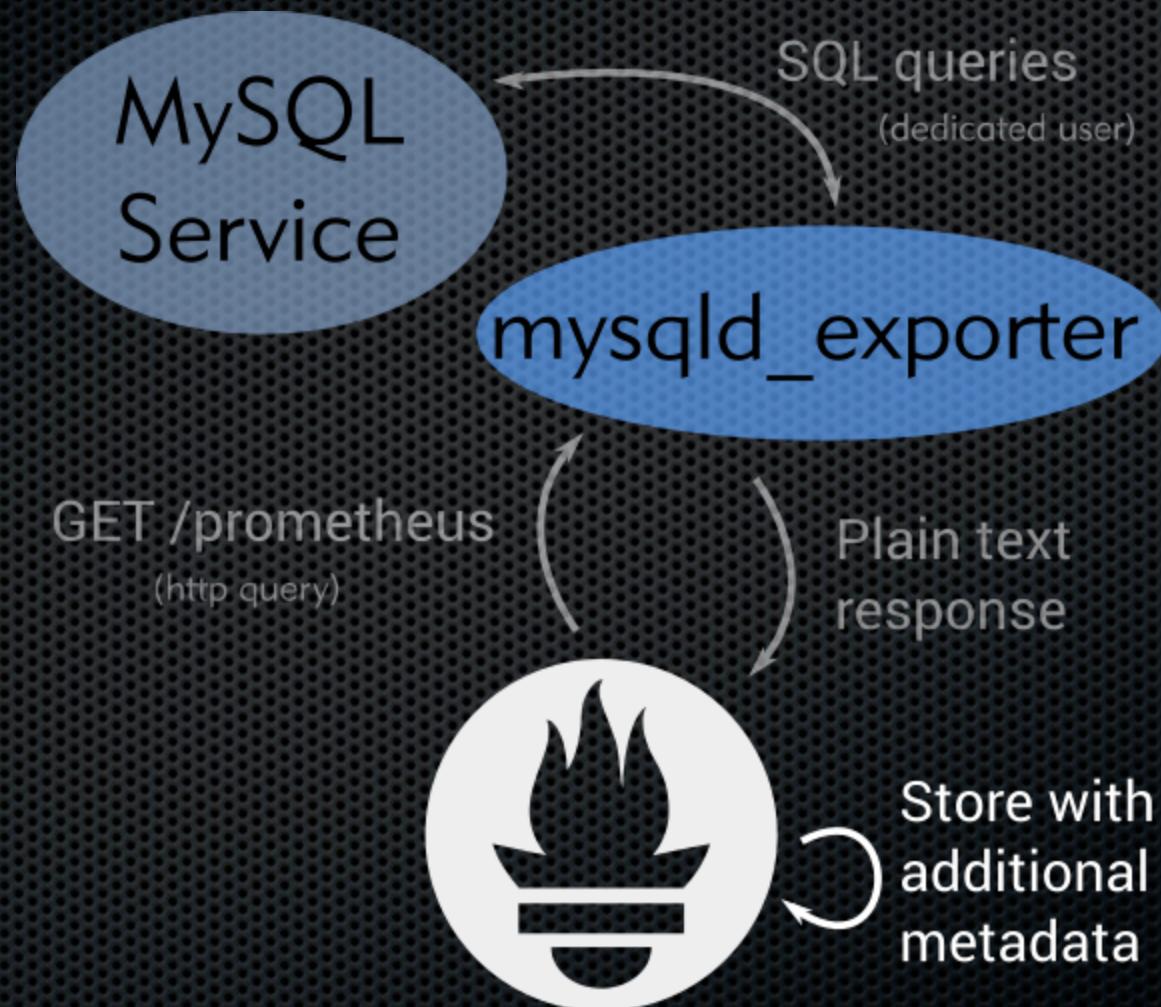
# How does it work?



# How does it work?



# How does it work?



# Exporters

- Exporters expose metrics with an HTTP API
- Bindings available for many languages
- Exporters do not save data ; they are not "proxies" and don't "cache" anything



# Exploring Metrics

Prometheus    Alerts    Graph    Status ▾    Help

Expression (press Shift+Enter for newlines)

Execute - insert metric at cursor ▾

Graph    Console

Element	Value
no data	

Remove Graph

Add Graph



# Exploring Metrics

Prometheus    Alerts    Graph    Status ▾    Help

Expression (press Shift+Enter for newlines)

Execute - insert metric at cursor -

Graph C Element no data Add Graph Remove Graph

- mysql\_global\_status\_innodb\_lsn\_current
- mysql\_global\_status\_innodb\_lsn\_flushed
- mysql\_global\_status\_innodb\_lsn\_last\_checkpoint
- mysql\_global\_status\_innodb\_master\_thread\_active\_loops
- mysql\_global\_status\_innodb\_master\_thread\_idle\_loops
- mysql\_global\_status\_innodb\_max\_trx\_id
- mysql\_global\_status\_innodb\_mem\_adaptive\_hash
- mysql\_global\_status\_innodb\_mem\_dictionary
- mysql\_global\_status\_innodb\_mem\_total
- mysql\_global\_status\_innodb\_mutex\_os\_waits
- mysql\_global\_status\_innodb\_mutex\_spin\_rounds
- mysql\_global\_status\_innodb\_mutex\_spin\_waits
- mysql\_global\_status\_innodb\_num\_index\_pages\_written
- mysql\_global\_status\_innodb\_num\_non\_index\_pages\_written
- mysql\_global\_status\_innodb\_num\_page\_compressed\_trim\_op
- mysql\_global\_status\_innodb\_num\_page\_compressed\_trim\_op\_saved
- mysql\_global\_status\_innodb\_num\_pages\_decrypted
- mysql\_global\_status\_innodb\_num\_pages\_encrypted
- mysql\_global\_status\_innodb\_num\_pages\_page\_compressed
- mysql\_global\_status\_innodb\_num\_pages\_page\_compression\_error

# Exploring Metrics

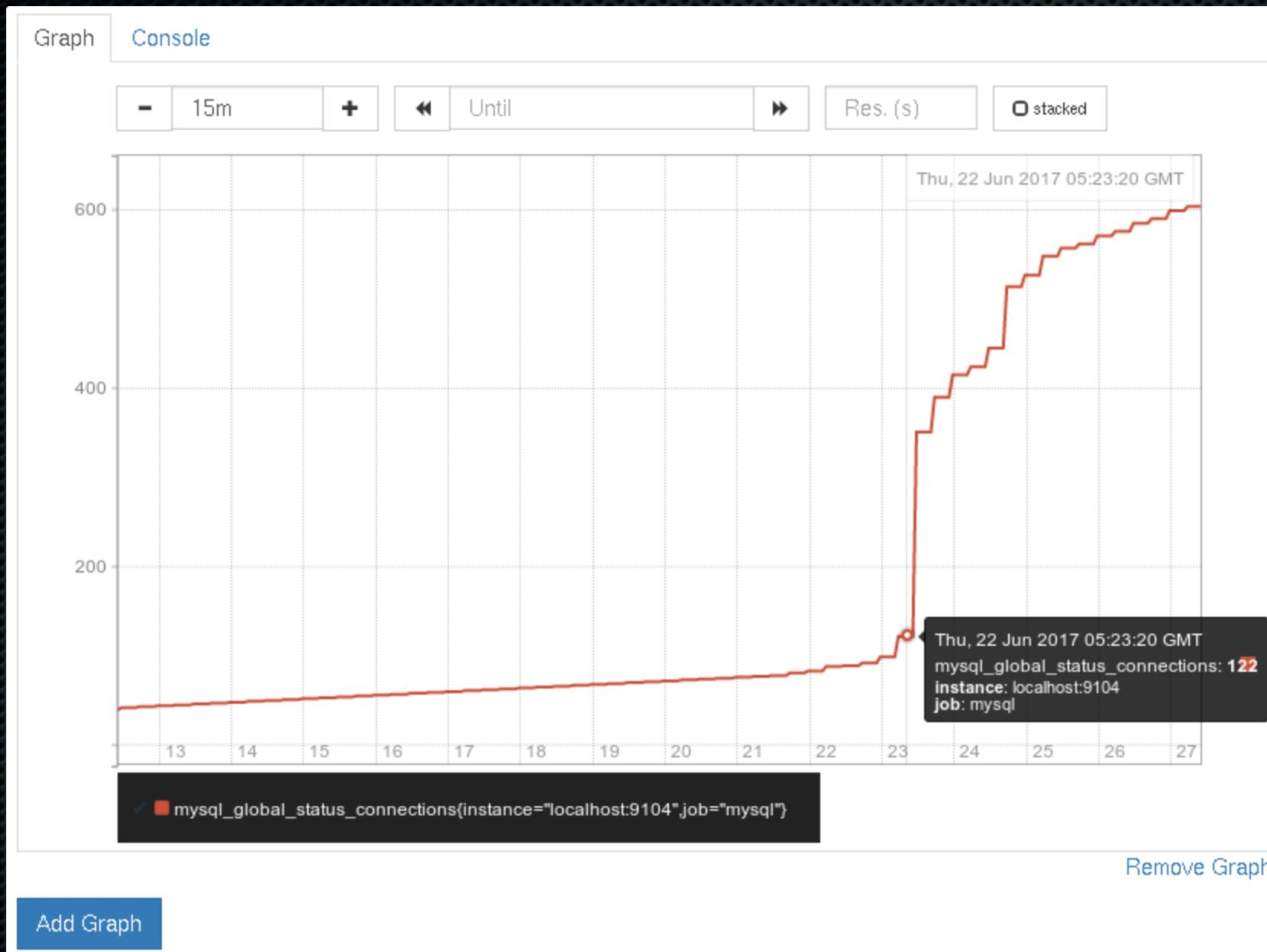
Prometheus    Alerts    Graph    Status ▾    Help

mysql\_global\_status\_connections

Execute  ▾ Load time: 38ms Resolution: 3s Total time series: 1

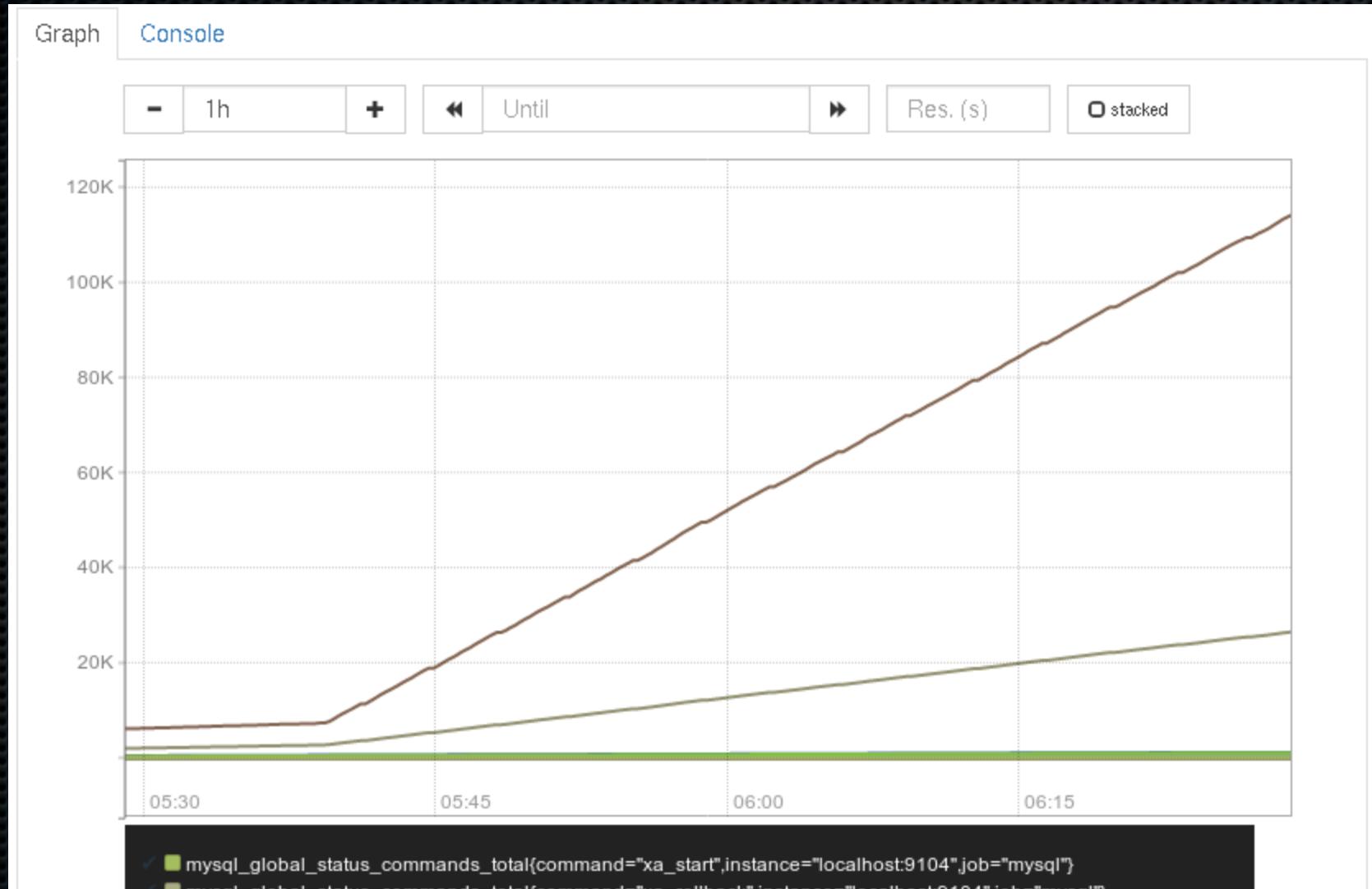


# Exploring Metrics



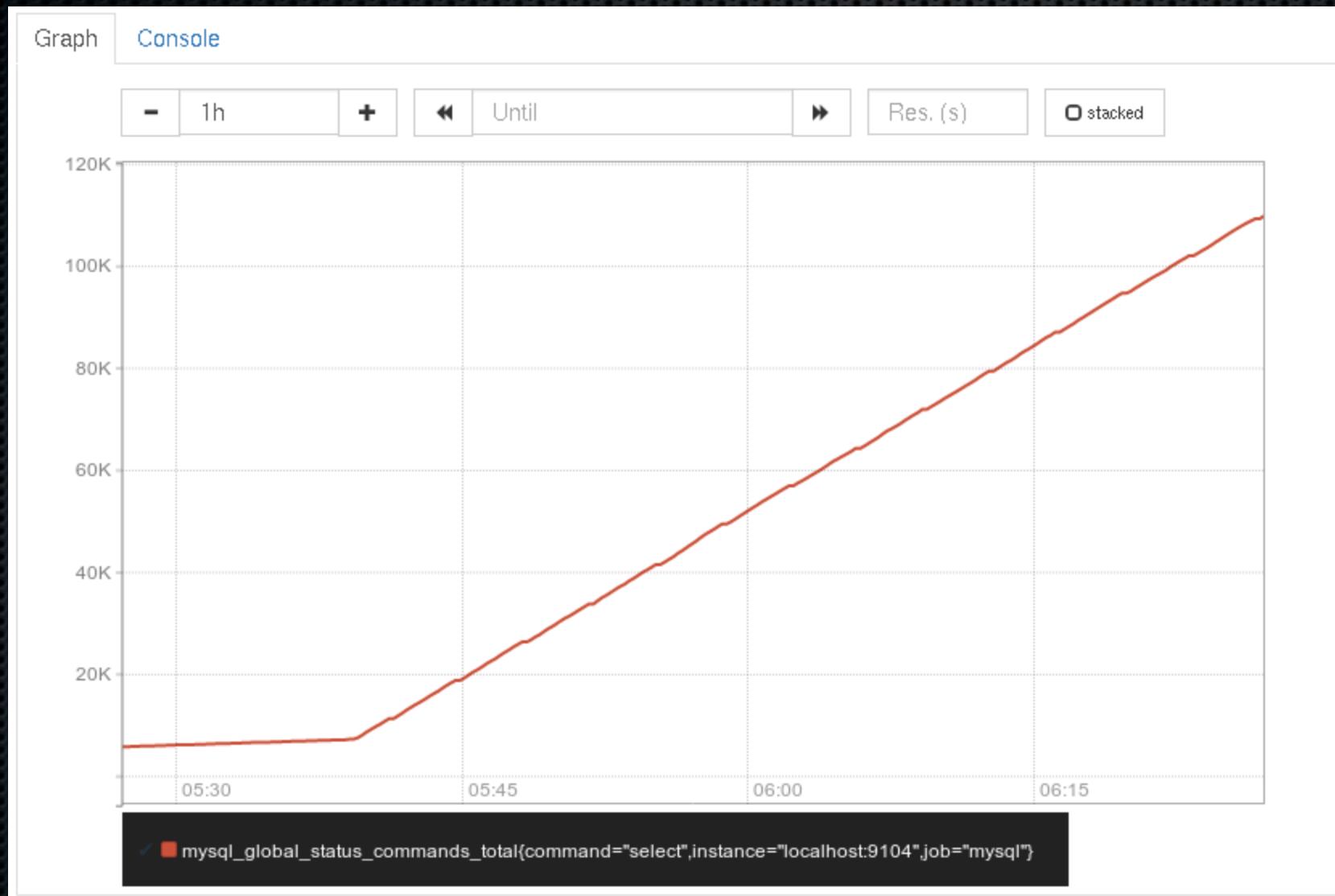
# PromQL

mysql\_global\_status\_commands\_total



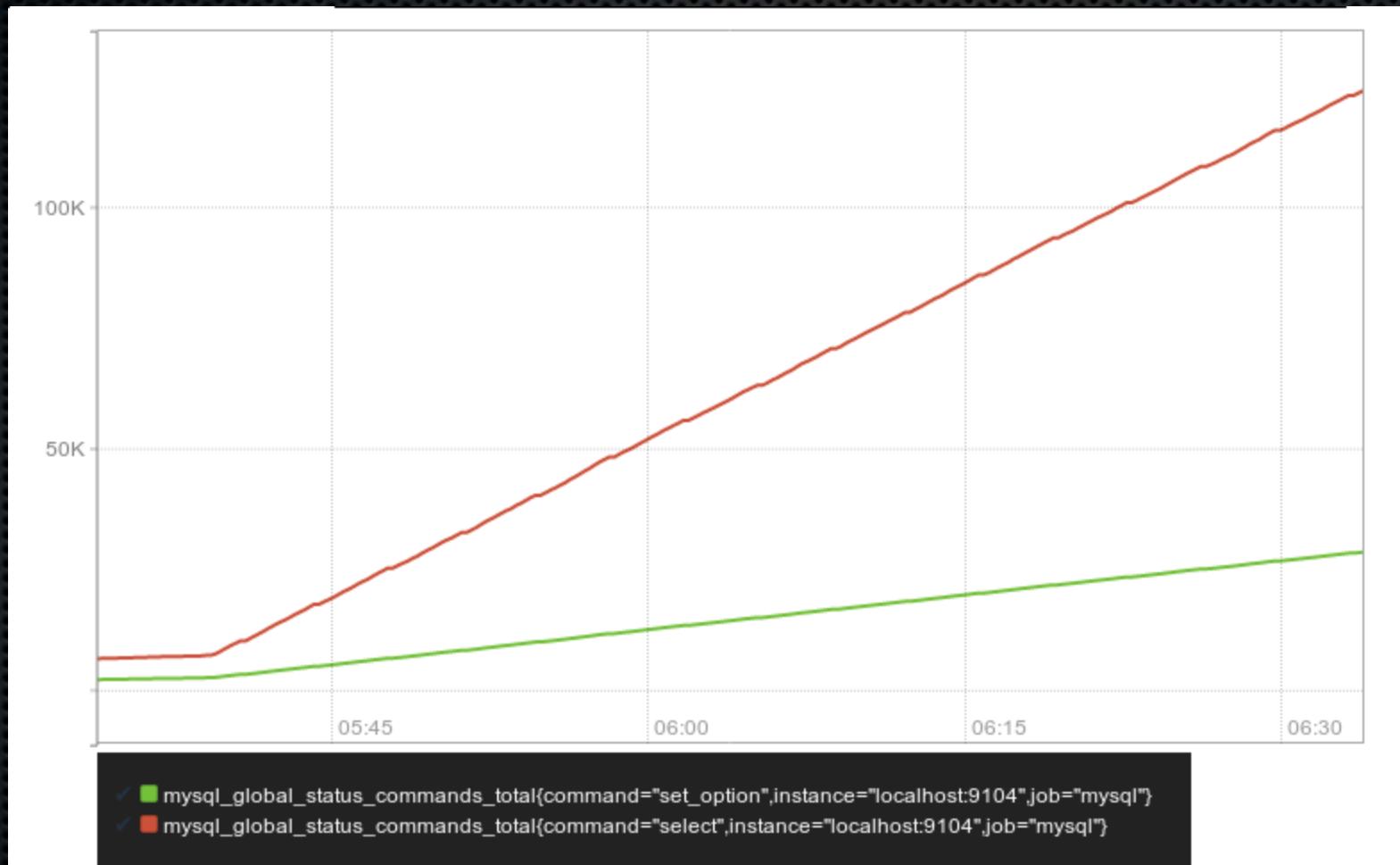
# PromQL

```
mysql_global_status_commands_total{command="select"}
```



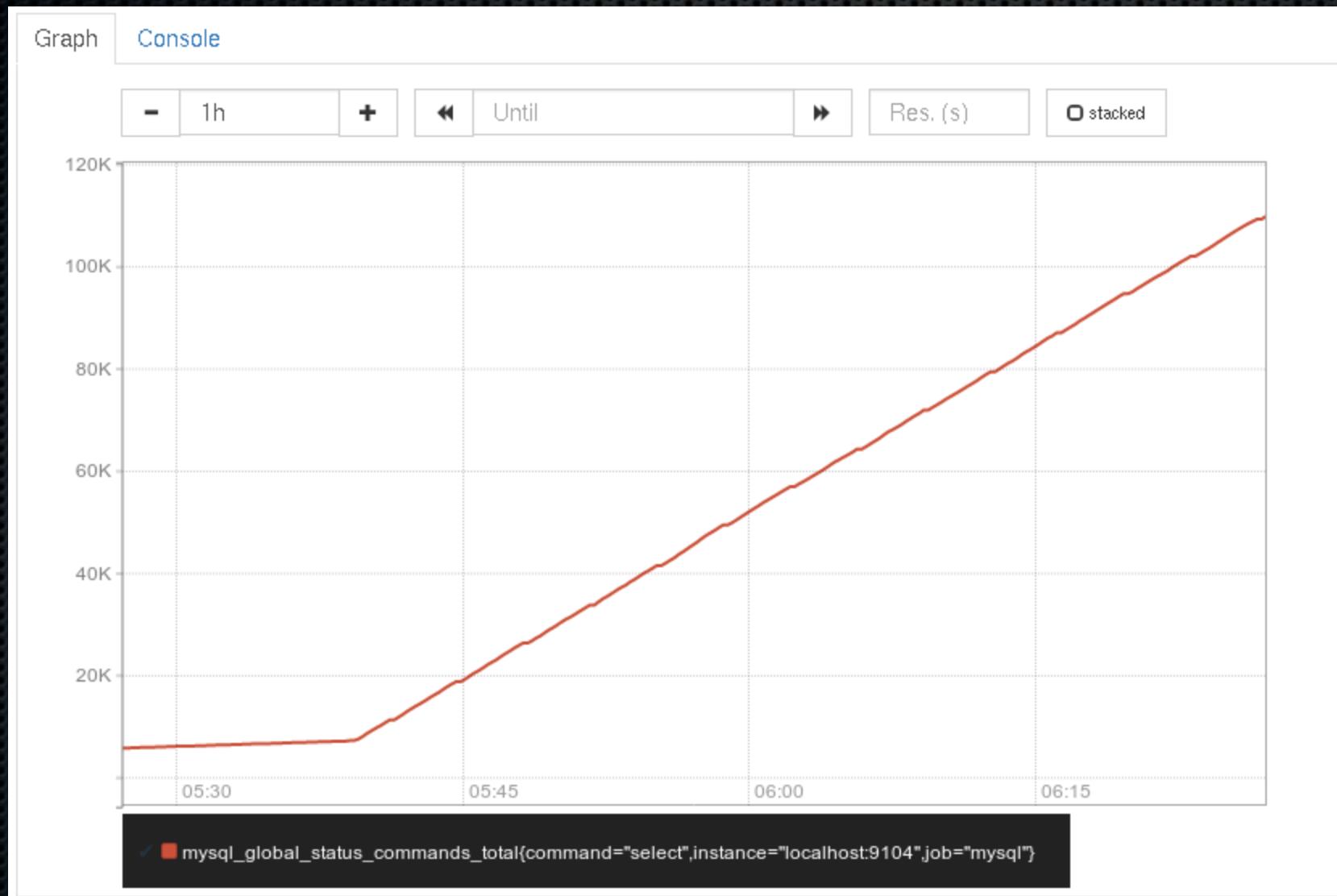
# PromQL

```
mysql_global_status_commands_total  
{command=~"select|set_options"}
```



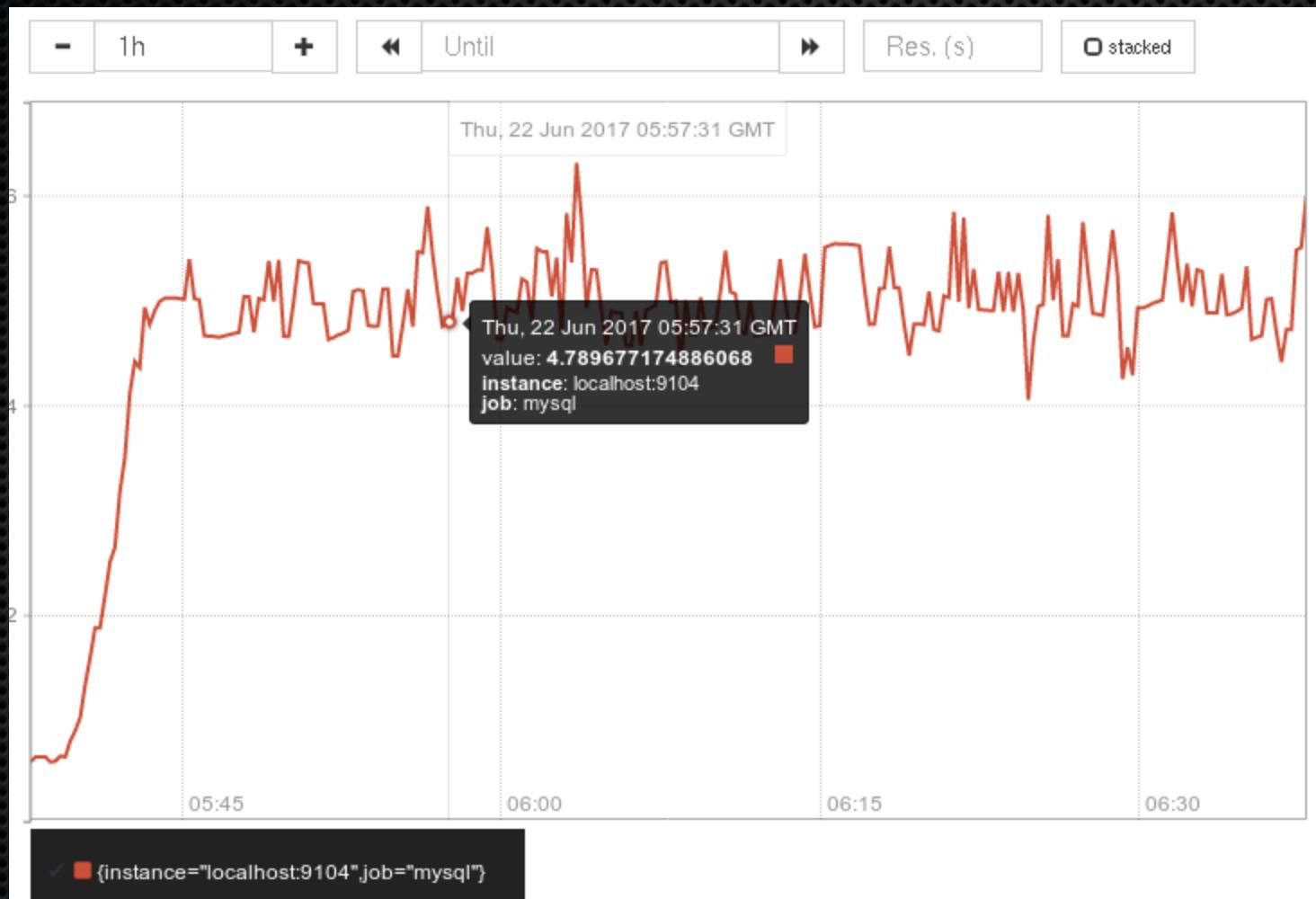
# PromQL

```
mysql_global_status_commands_total{command=~"select|set  
_options"}
```



# PromQL

deriv(mysql\_global\_status\_connections [5m] )



# PromQL

```
{__name__ =~ ".+innodb.+cache.*"}
```

```
predict_linear(mysql_heartbeat_lag_seconds[5m], 60*2)
```

```
sum(rate(mysql_global_status_commands_total{command=~"(commit|rollback)"} [5m])) without (command)
```



# Prometheus + mysql\_exporter

- Collect a LOT of data  
(e.g. global variables & status)
- Embeds basic visualisation console
- By default scraps every 15s



# A word about Prometheus vs Graphite

Prometheus does not see a metric as an "event".

Metrics are current value until they are replaced.

You can not see when a metric has been included in Prometheus.

*For Events, Prometheus refers to Elasticsearch.*





# Grafana



# One tool does one job...

- Prometheus will collect data
- Exporters will expose data
- Grafana will graph data



# Grafana

- Open Source (Apache 2.0)
- Web app
- Specialized in visualization
- Pluggable
- Multiple datasources: prometheus, graphite, influxdb...
- Has an API!



# History of Grafana

Grafana is a fork of Kibana 3 ; used to be JS-Driven.

Now fully featured, requires a database, multi-projects/users support, etc...

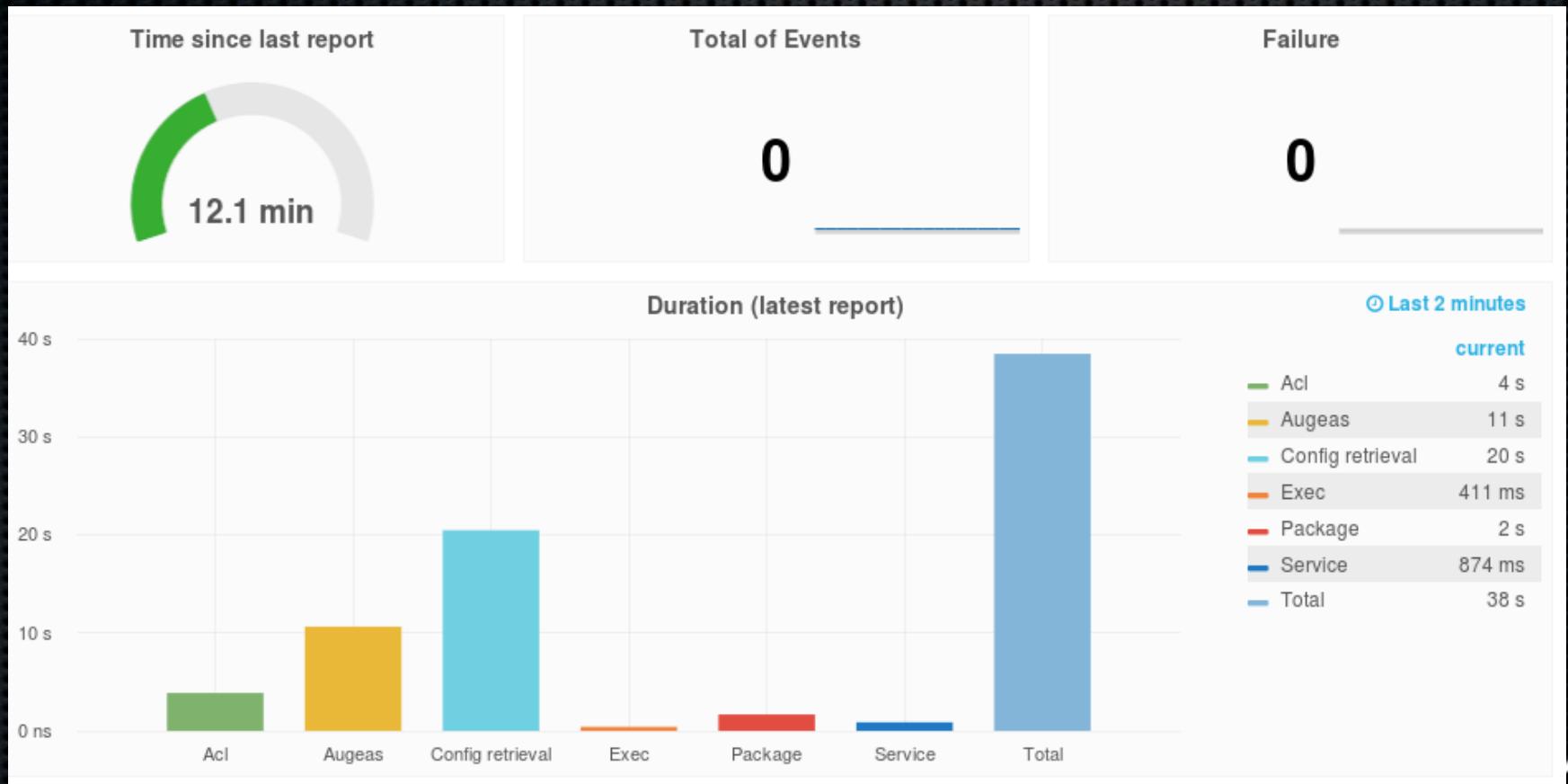


# Grafana and Prometheus

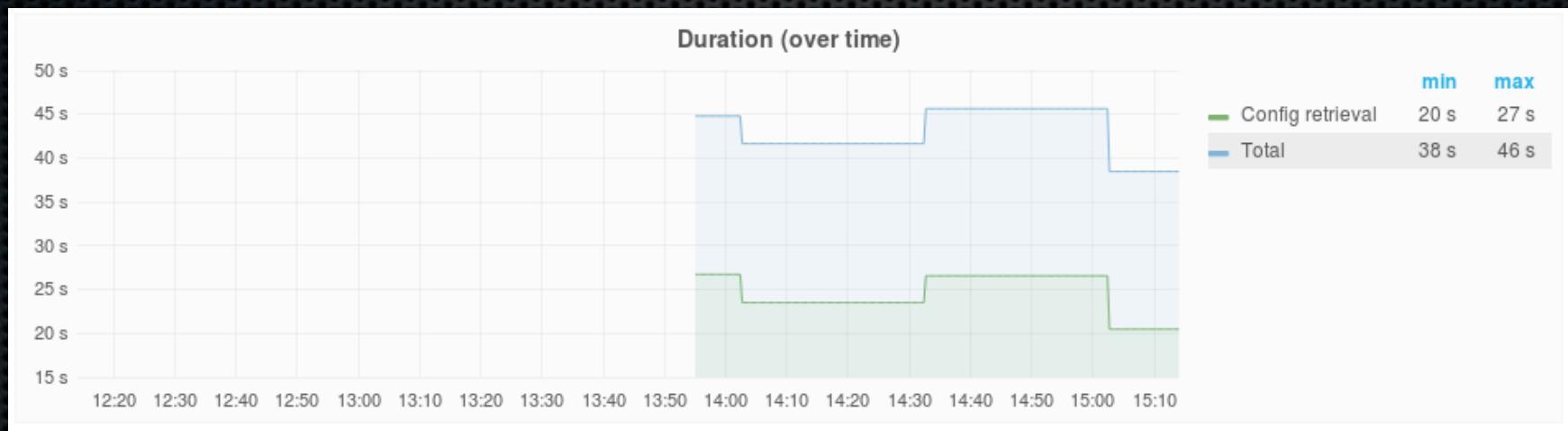
- Prometheus shipped its own consoles
- Now it recommends Grafana and deprecated its own consoles



# Grafana Dashboards



# Grafana Dashboards



# Time Picker

Time range

From: now-12h

To: now

Refreshing every: off

Quick ranges			
Last 2 days	Yesterday	Today	Last 5 minutes
Last 7 days	Day before yesterday	Today so far	Last 15 minutes
Last 30 days	This day last week	This week	Last 30 minutes
Last 90 days	Previous week	This week so far	Last 1 hour
Last 6 months	Previous month	This month	Last 3 hours
Last 1 year	Previous year	This month so far	Last 6 hours
Last 2 years		This year	<b>Last 12 hours</b>
Last 5 years		This year so far	Last 24 hours

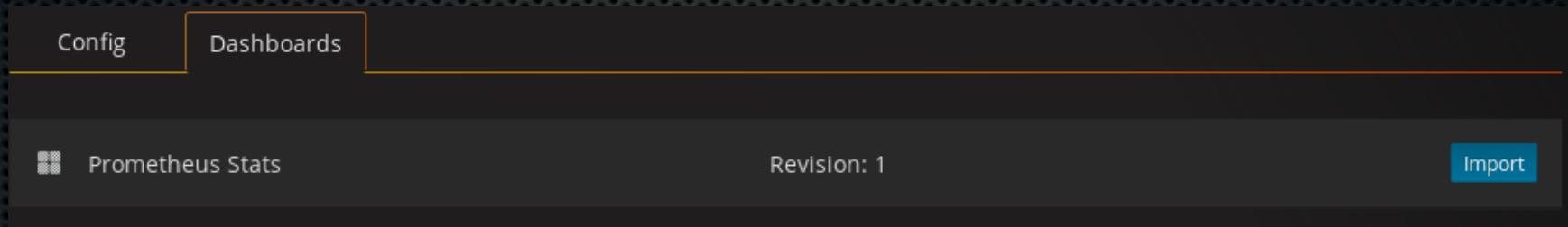


# Configure Prometheus in Grafana

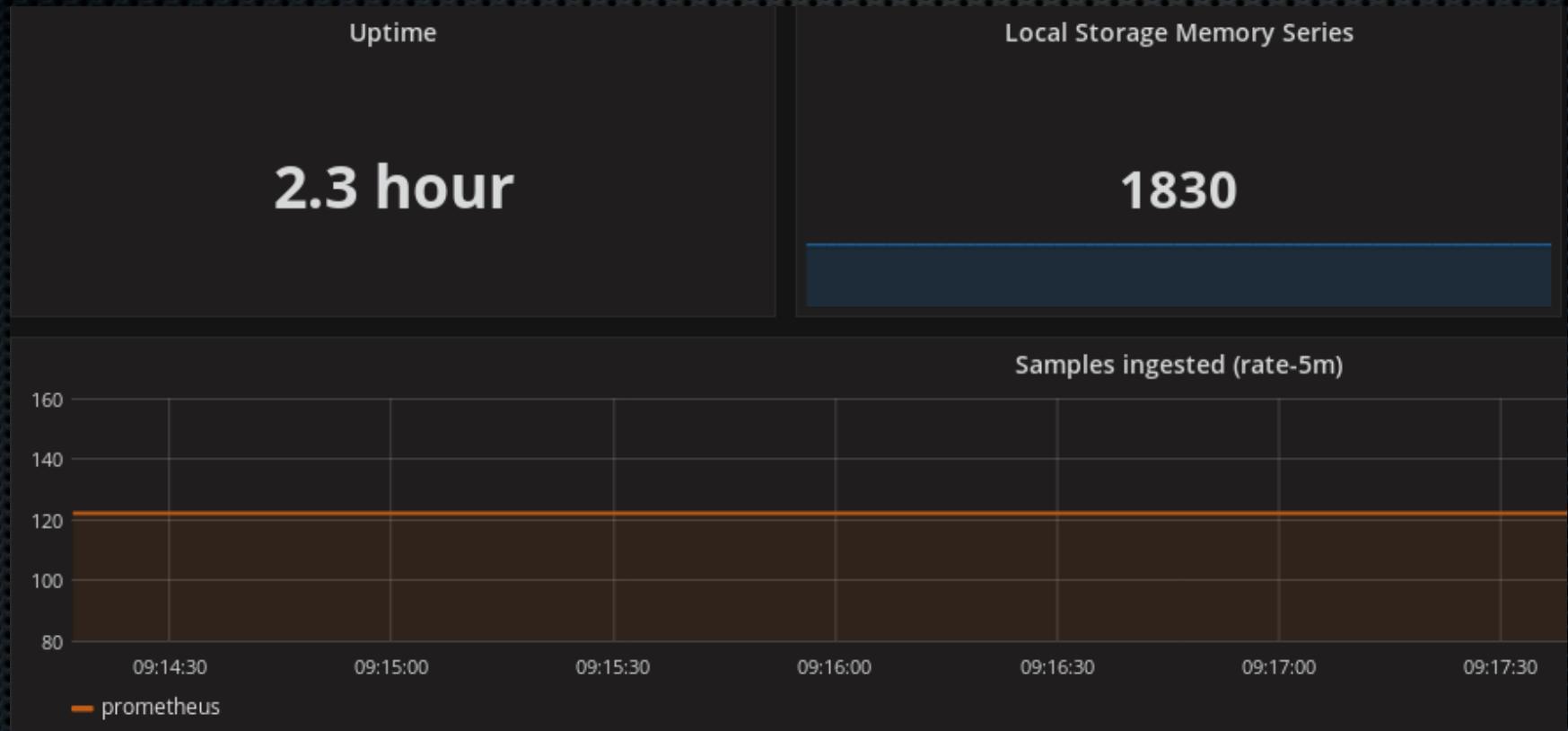
Name	Prometheus	<a href="#">i</a>	Default	<input checked="" type="checkbox"/>
Type	Prometheus	<a href="#">▼</a>		
Http settings				
Url	http://localhost:9090	<a href="#">i</a>		
Access	proxy	<a href="#">▼</a>	<a href="#">i</a>	
Http Auth				
Basic Auth	<input type="checkbox"/>	With Credentials	<a href="#">i</a>	<input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/>	With CA Cert	<a href="#">i</a>	<input type="checkbox"/>



# Configure Prometheus in Grafana



# Prometheus Dashboard



# Creating Grafana Dashboards

- Takes time
- Requires deep knowledge of the tools
- Improved over time
- Easy to share (json + online library)



# Percona Grafana Dashboard

- Percona Open Sourced Grafana Dashboards
- Covering MySQL, Mongo **and** Linux monitoring
- Part of a bigger picture, PMM, but usable standalone
- Open Source (AGPL!)
- <https://github.com/percona/grafana-dashboards>



# Installing Percona Graphes

## Method 1 (RO dashboards)

- Enable File dashboards in Grafana
- Clone grafana-dashboards to the configured location (or make a package)

## Method 2 (RW dashboards)

- Use the Grafana API to upload the JSON's.



# MySQL Setup

- You'll need mysqld\_exporter, with a user
- MySQL 5.1+
- Performance Schema for full set of metrics
- mysqld\_exporter
  - collect.binlog\_size=true
  - collect.info\_schema.processlist=true`



# node\_exporter setup

- node\_exporter  
-collectors.enabled=  
"diskstats,filefd,filesystem,loadavg,  
meminfo,netdev,stat,time,uname,vmstat"



# Prometheus (static file)

```
scrape_configs:
  - job_name: prometheus
    static_configs:
      - targets: ['localhost:9090']
        labels:
          instance: prometheus

  - job_name: linux
    static_configs:
      - targets: ['10.0.98.43:9100']
        labels:
          instance: db1

  - job_name: mysql
    static_configs:
      - targets: ['10.0.98.43:9104']
        labels:
          instance: db1
```



# Dashboards

■■■ Amazon RDS OS Metrics	Cloud	Percona	☆
■■■ Cross Server Graphs	Insight	Percona	☆
■■■ Disk Performance	OS	Percona	☆
■■■ Disk Space	OS	Percona	☆
■■■ MariaDB	MySQL	Percona	☆
■■■ MongoDB Cluster Summary	MongoDB	Percona	☆
■■■ MongoDB InMemory	MongoDB	Percona	☆
■■■ MongoDB MMAPv1	MongoDB	Percona	☆
■■■ MongoDB Overview	MongoDB	Percona	☆
■■■ MongoDB ReplSet	MongoDB	Percona	☆
■■■ MongoDB RocksDB	MongoDB	Percona	☆



# Dashboards

MongoDB WiredTiger	MongoDB	Percona	☆
MySQL InnoDB Metrics	MySQL	Percona	☆
MySQL InnoDB Metrics Advanced	MySQL	Percona	☆
MySQL MyISAM Metrics	MySQL	Percona	☆
MySQL Overview	MySQL	Percona	☆
MySQL Performance Schema	MySQL	Percona	☆
MySQL Query Response Time	MySQL	Percona	☆
MySQL Replication	MySQL	Percona	☆
MySQL Table Statistics	MySQL	Percona	☆
MySQL TokuDB Metrics	MySQL	Percona	☆
MySQL User Statistics	MySQL	Percona	☆



# Dashboards

PXC/Galera Cluster Overview	HA	Percona	☆
PXC/Galera Graphs	HA	Percona	☆
Prometheus		Percona	☆
Prometheus Stats		prometheus	☆
ProxySQL Overview	HA	Percona	☆
Summary Dashboard	Insight	Percona	☆
System Overview	OS	Percona	☆
Trends Dashboard	Insight	Percona	☆

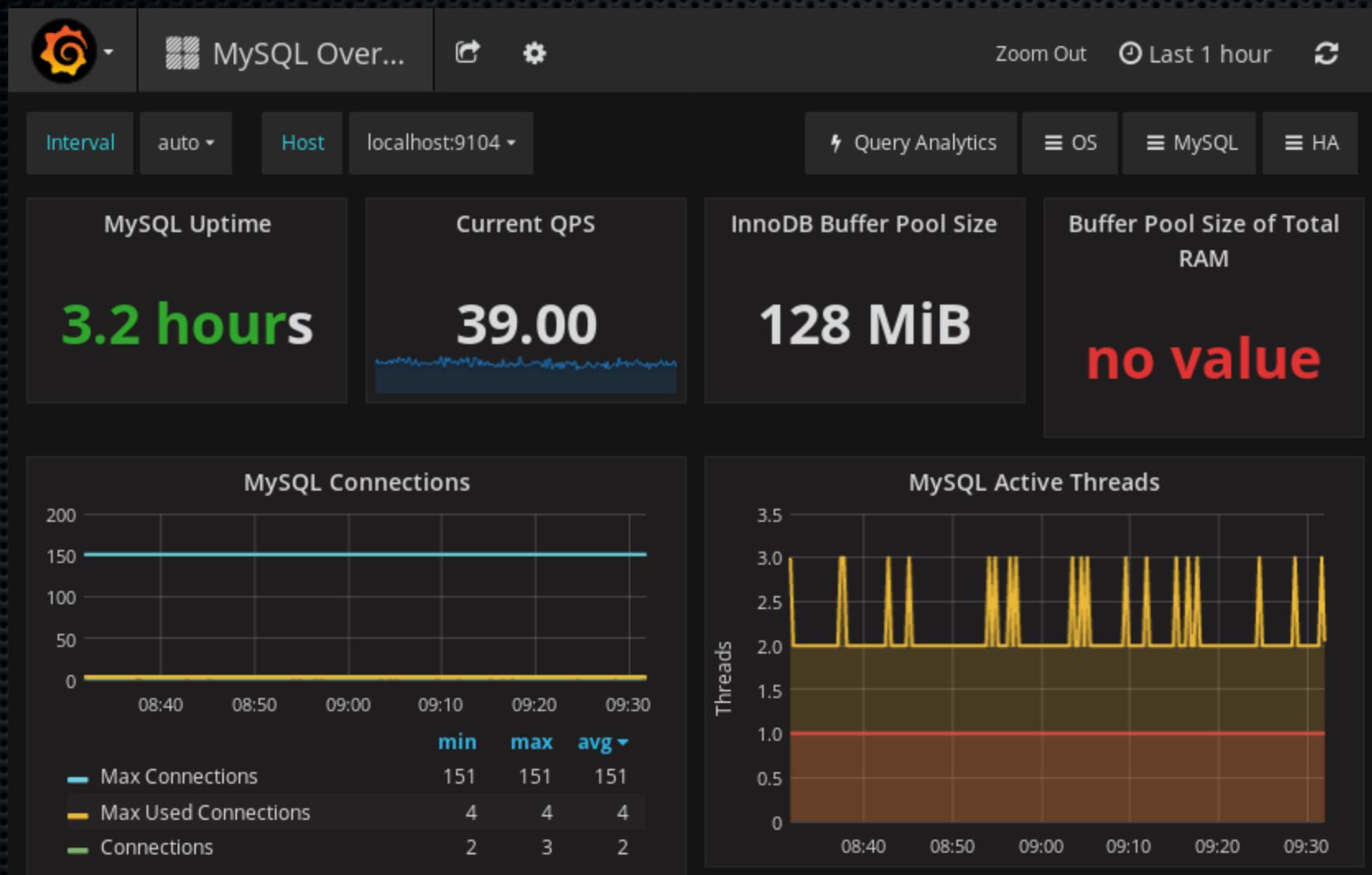


# We don't need all of them?

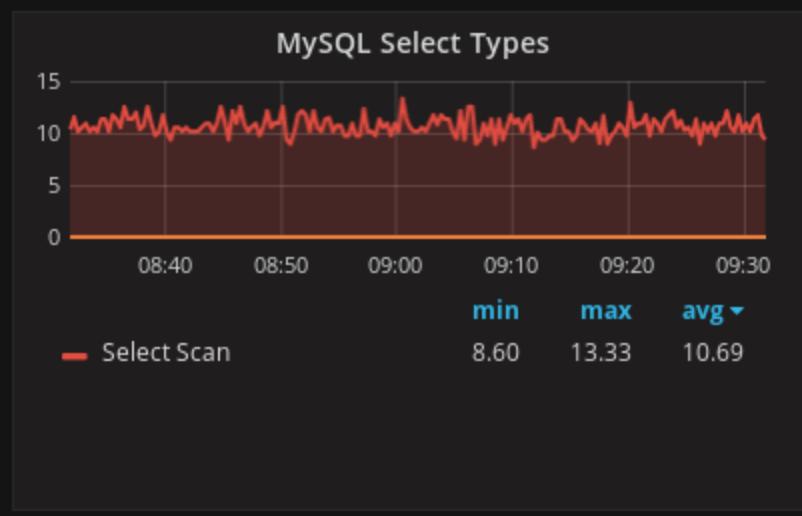
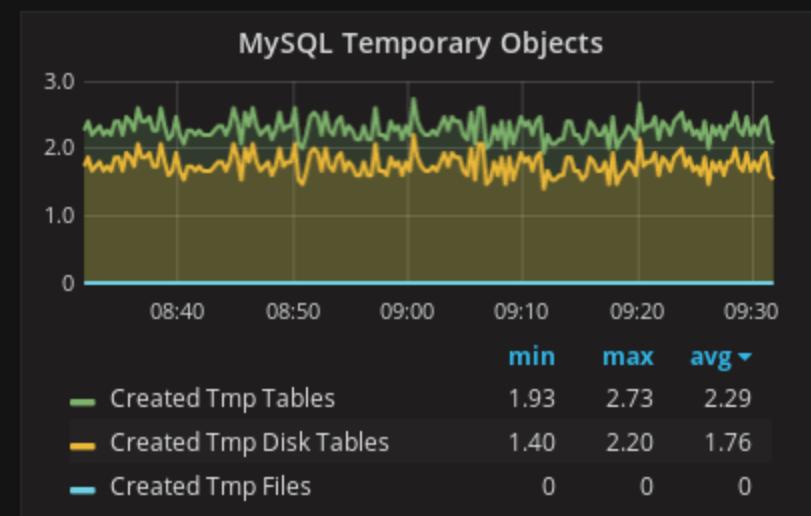
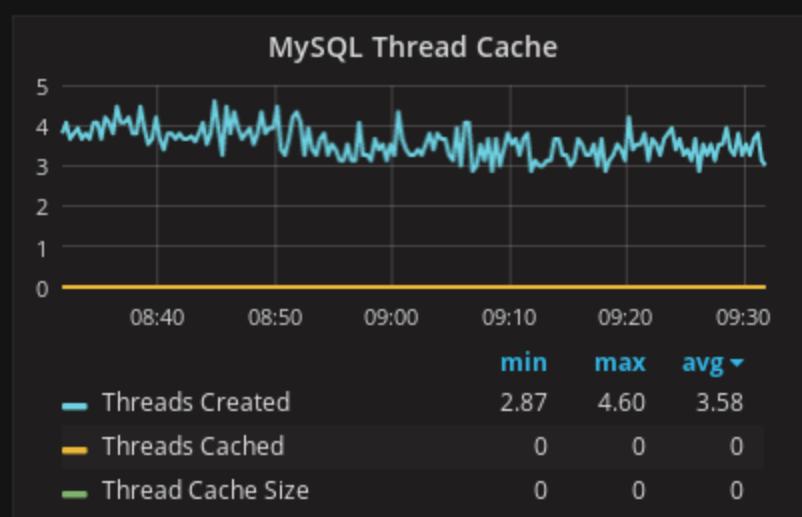
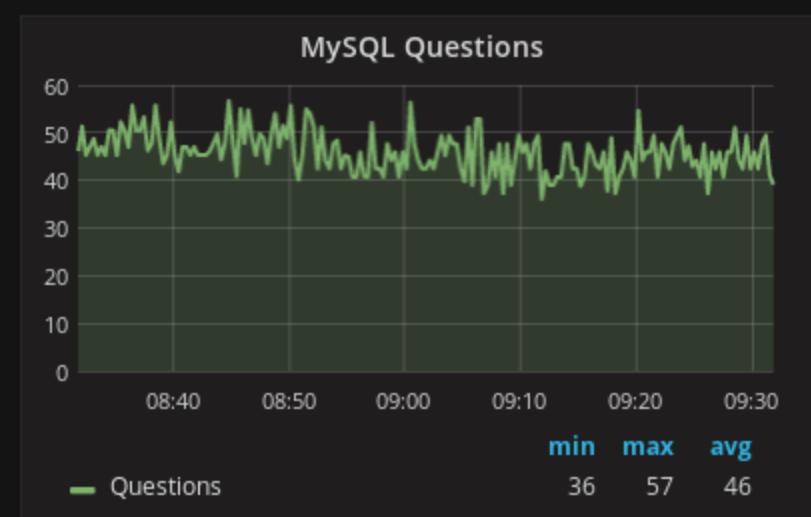
- Because Grafana is just viz, you can import only the one you want (e.g. exclude Mongo)
- You can import later any extra dashboard you need



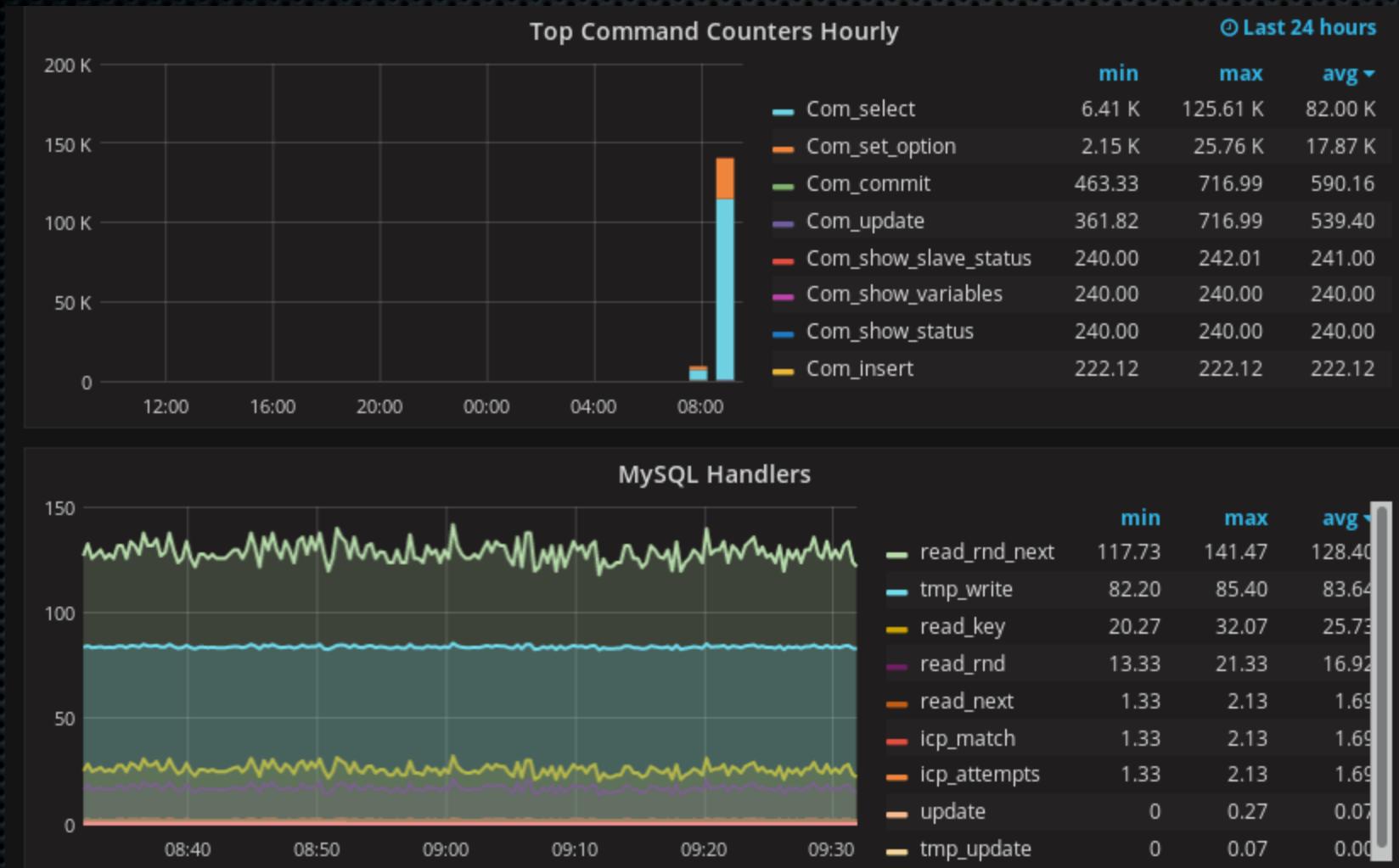
# MySQL Overview



# MySQL Overview



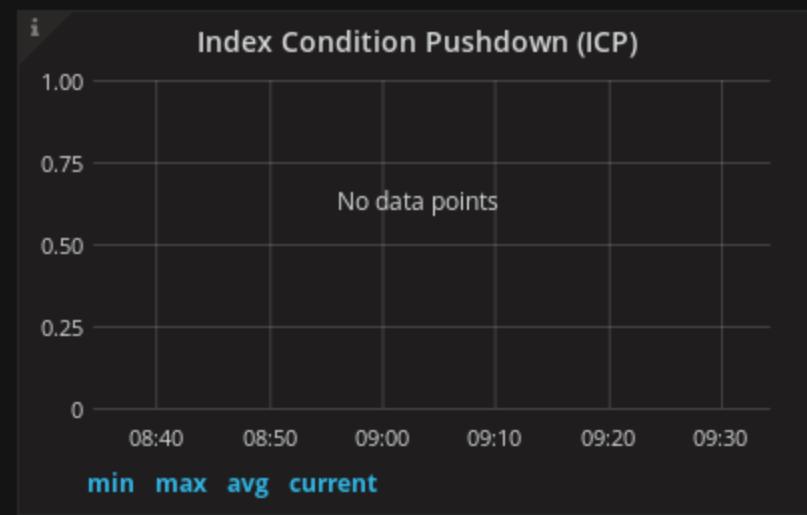
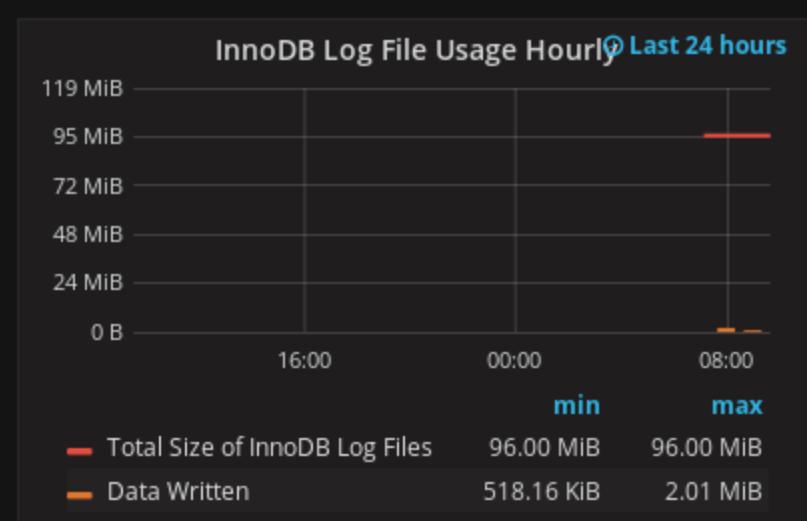
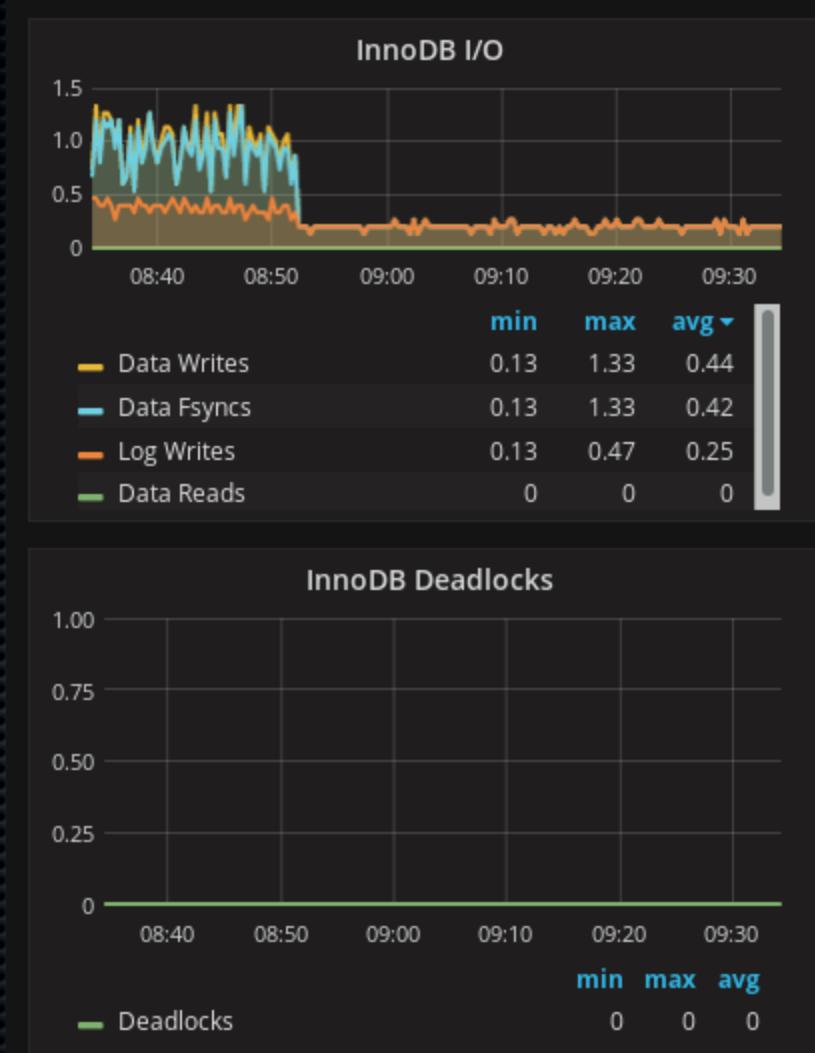
# MySQL Overview



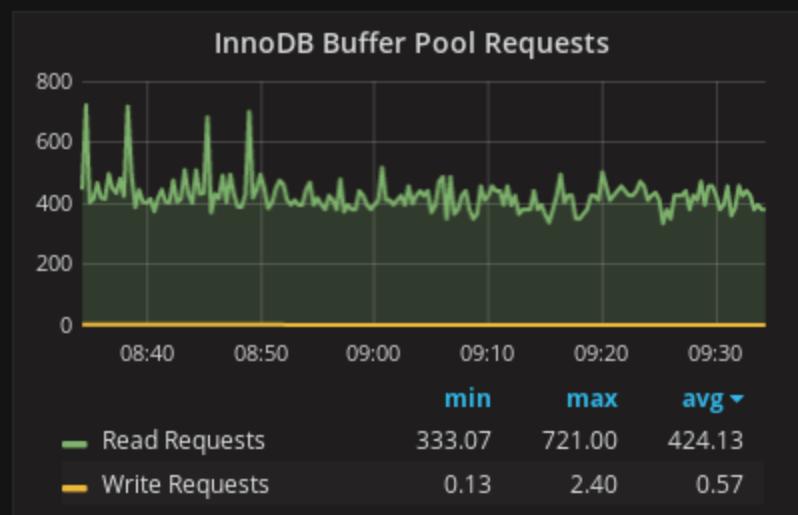
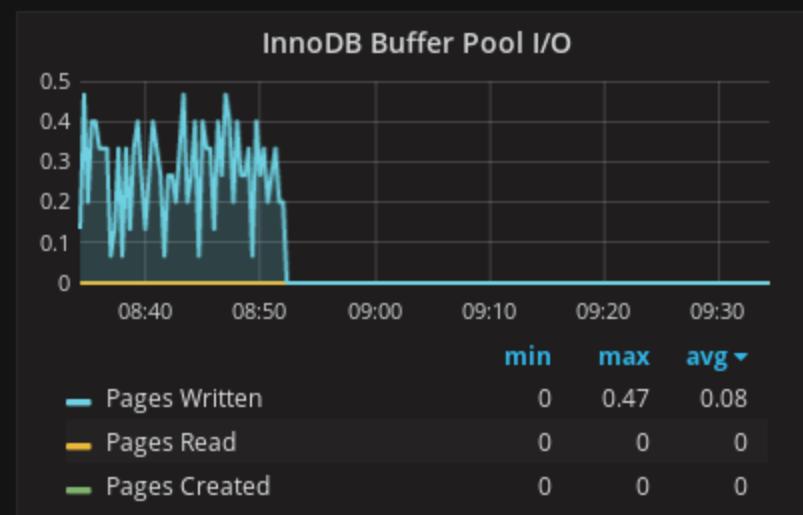
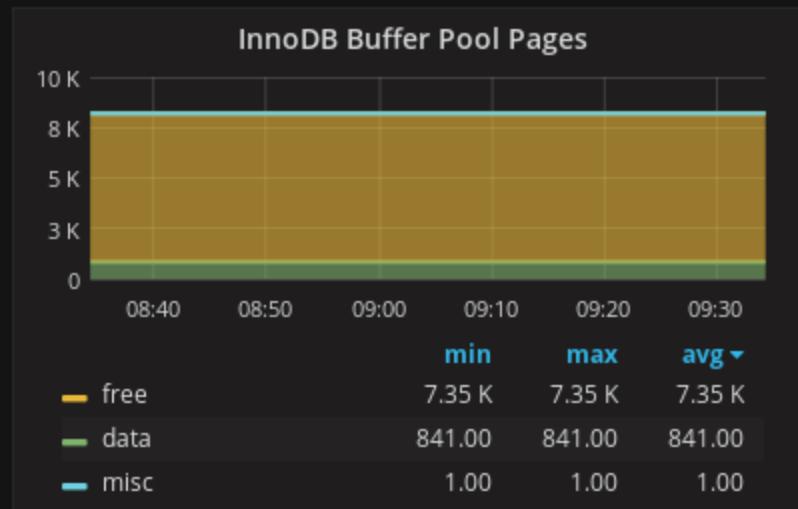
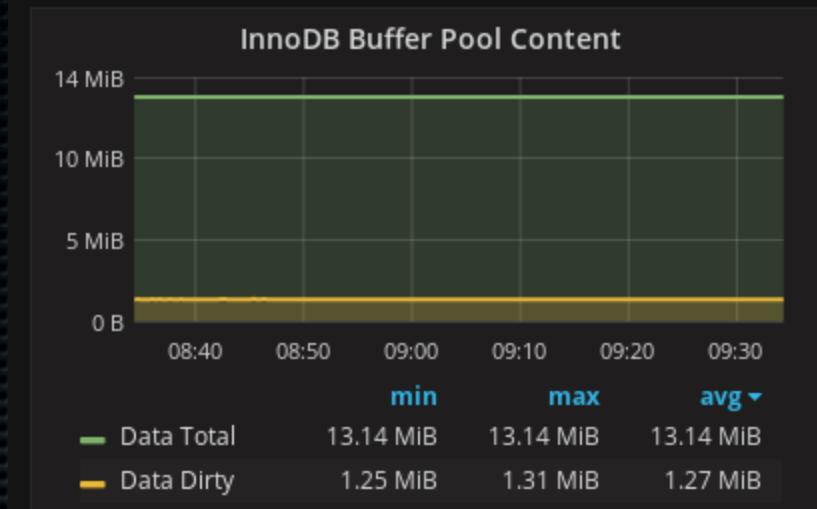
# InnoDB



# InnoDB



# InnoDB



# Replication

IO Thread Running

Yes

SQL Thread Running

Yes

MySQL Replication Delay



# Conclusions

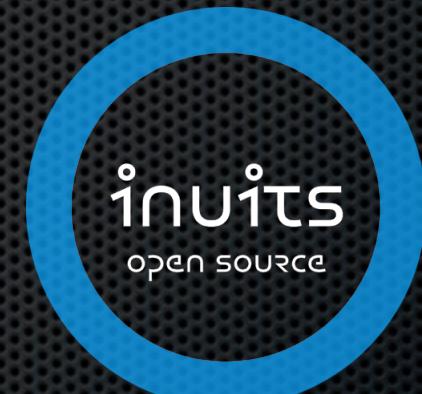
- Prometheus and Grafana are first-class monitoring tools
- Totally different approach than other tools
- Embeddable into your apps
- Percona Dashboards gets your graphs ready in no-time with minimal efforts



# Contact



Julien Pivotto  
*roidelapluie*  
roidelapluie@inuits.eu



Inuits  
<https://inuits.eu>  
info@inuits.eu

